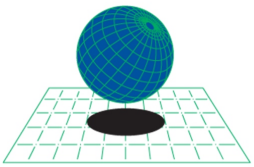


CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Infraestructura de càlcul HPC CIMNE

Autor:	Felip Moll Marquès
Director:	Miguel A. Pasenau de Riera
Ponent:	Fermín Sanchez Carracedo
Departament del ponent:	Arquitectura de Computadors
Titulació:	Enginyeria Informàtica superior
Centre:	Facultat d'Informàtica de Barcelona (FIB)
Universitat:	Universitat Politècnica de Catalunya (UPC)
Empresa:	CIMNE
Data:	Abril 2011 – Octubre 2012

Copyright

Agraïments

La realització d'aquest projecte no hagués estat possible sense la col·laboració de les següents persones:

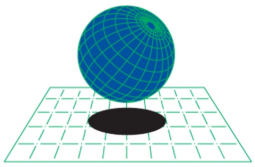
Personal de CIMNE
Miguel Alonso
Miguel Pasenau
Pooyan Dadvand
Ricardo Rossi
Jordi Cotela

Família i Amics
Salo Moll
Elena Marquès
Núria Seguí

Sobre traduccions...

La llengua d'aquest document és el Català, no obstant s'ha decidit mantenir la terminologia tècnica en l'idioma anglès en la majoria dels casos amb l'únic objectiu de facilitar la cerca de contingut en les referències de la bibliografia. Com a exemple, un “chassis”, “switch” o un “rack” no es traduiran a “xassís”, “encaminador” o “armari” amb l'únic objectiu de no confondre el lector i facilitar-li la comprensió i cerca.

Continguts



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 1
Introducció

1.1 Motivació

CIMNE, el Centre Internacional de Mètodes Numèrics en Enginyeria és un centre dedicat a la investigació en diversos camps i especialment en mètodes numèrics. En aquest centre hi participen investigadors que tenen unes necessitats de càlcul computacional elevades.

Vaig començar a treballar al CIMNE l'any 2009. En aquell moment se'm van encarregar certes tasques de re-estructuració del centre de càlcul com la substitució del servidor de fitxers i el de correu. Més endavant vaig anar agafant més responsabilitats i experiència.

Des de 2008 CIMNE comptava amb una infraestructura de càlcul dotada de dos servidors i un clúster per donar als investigadors la potència de càlcul necessària. Fins el moment, el servei havia estat mal utilitzat degut a les deficiències d'instal·lació i planificació, a la falta de documentació tant d'usuari com d'administrador i als continus i diversos problemes que sofria la infraestructura. A més, el desconeixement de l'equip de sistemes en quant al servei de càlcul era notori i també les demandes que els usuaris feien al departament.

El servei de càlcul i en especial el clúster de CIMNE era la bèstia gran, el projecte que ningú s'atrevia a tocar i en definitiva un gran repte. No tenia cap coneixement del que en realitat suposava la gestió i la implementació d'un clúster de càlcul, ni de com estava estructurat, ni de com es planificaven els treballs i les cues, ni de que era el famós Infiniband, etc. Em semblava una tasca realment complicada. A més, els preus que ens demanaven per re-estructurar el servei eren per jo desorbitats i em feia pensar que no era una labor qualsevol.

Aquests motius són els que em van fer decidir. No podia pensar en ser responsable d'un servei que fos una caixa negra de la que no en tinguéssim el control, era intolerable no poder respondre als problemes dels usuaris. Necessitava saber com funcionava un clúster fins al nivell de poder-lo implementar. Per altra banda no podia pensar que hi hagués una tasca tant complicada que no la pogués complir en aquell entorn i de fet aquest projecte es convertí en una prova de superació personal: si era capaç de fer això, seria capaç de fer qualsevol cosa a l'empresa. Així va ser, el projecte l'he desenvolupat amb moltes hores de dedicació, però també amb moltes ganes i pretenent en tot moment assolir l'excel·lència en cada tasca realitzada. Realment he de dir que ha tingut recompensa i estic orgullós d'aquest projecte.

Per què un PFC?

En definitiva el projecte pretén satisfer les necessitats computacionals del centre i és per això que es necessita un projecte d'enginyeria per tal de tenir en compte totes les variables que envolten aquest i solucionar i satisfer de forma eficaç tots els problemes i requisits que en derivin.

És important mantenir funcionant el servei durant l'etapa de desplegament ja que cada aturada es tradueix en una disminució de la productivitat dels investigadors. A tal efecte s'haurà de portar a terme una planificació metodològica que no interrompi, en la mesura que sigui possible, els treballs actuals en execució i que sigui el més transparent possible per l'usuari.

Per integrar l'actual sistema en el centre de processament de dades de CIMNE es requeriran coneixements de xarxes i administració de sistemes. Per els anàlisis del hardware i de consum necessitarem coneixements en arquitectura i estructura de computadors. Per la realització de proves s'haurà de tenir clar el concepte de paral·lelisme i tenir suficient habilitat per escriure algorismes que l'explotin. Òbviament també es necessitarà una base sòlida de sistemes operatius, en especial GNU/Linux i la seva administració.

Tots aquests coneixements són impartits en diverses assignatures a la carrera d'Enginyeria Informàtica i fan que no sigui possible o recomanable realitzar aquest projecte per personal amb estudis no superiors que no hagin rebut la formació adequada.

El projecte es realitzarà com a PFC ja que coincideix amb els requisits que es demanen en relació al nombre d'hores d'execució i de complexitat. A més s'haurà de realitzar la investigació sobre tots els temes que no es coneguin, partint de la base que gairebé no es té noció del funcionament real d'un clúster de càlcul com el disponible al CIMNE.

1.2 Objectius

L'objectiu d'aquest projecte és el de renovar integralment la infraestructura de càlcul del CIMNE determinant quins dels equipaments actuals es poden utilitzar, llavors implementant un nou sistema operatiu al clúster, aplicant un gestor de recursos i planificador de treballs centralitzat i eficaç, proporcionant una documentació completa tant pels usuaris com pels administradors juntament amb procediments estàndard de gestió, millorant la seguretat de tot el conjunt, implementant sistemes de monitorització, determinant alguna plataforma de comunicació i documentació entre usuaris i realitzant dos estudis, un referent a l'arquitectura del hardware i l'altre de consum i sostenibilitat, aspecte que no s'havia tingut en compte fins el plantejament d'aquest projecte. També es pretenen satisfer altres possibles requisits que vagin sorgint durant la realització del treball.

En definitiva i resumint, es pretén implementar un servei de càlcul complet per el CIMNE.

1.3 Organització de la memòria

Fins aquí hem pogut veure el primer capítol on hem explicat en que consisteix aquest projecte d'enginyeria, quins en són els objectius i perquè s'ha de portar a terme.

A continuació entrarem al capítol 2 on farem un anàlisis complet del CIMNE i la actual infraestructura de càlcul. Veurem quins són els principals usos i problemes que existeixen i n'obtindrem uns requisits funcionals que determinaran les tasques més concretes a realitzar.

Amb aquests requisits passarem al capítol 3 on farem una planificació temporal i un anàlisis de costos respecte el que podria suposar aquest projecte. Addicionalment explicarem quines eines de gestió farem servir per portar a terme el projecte.

A continuació trobarem el capítol 4 on realitzarem una investigació en profunditat de tot el hardware del que disposem al centre de processament de dades. En base a aquest estudi previ determinarem l'estat de l'art en aquest camp i quines són les alternatives actuals per implementar una solució que compleixi els objectius.

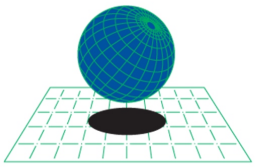
Un cop definida la solució, en el capítol 5 detallarem els passos per realitzar una primera implementació que verificarem i validarem aplicant correccions si és necessari.

A partir del capítol 6 analitzarem com s'ha portat a terme el desplegament de la solució a tots els nodes de càlcul. També mostrarem quins problemes hi ha hagut i com hem validat la instal·lació.

Un cop desplegat tot el sistema i en funcionament implementarem en el capítol 7 les eines de suport i documentació per els administradors i els usuaris. Com a extensió d'aquesta documentació entrarem al capítol 8 on realitzarem els estudis tècnics de sostenibilitat i arquitectura del hardware.

Acabarem amb el capítol 9 amb les conclusions del projecte avaluant si s'han complert els objectius, la planificació, els costos i determinant el treball futur.

Al final de la memòria es podrà trobar la bibliografia i diversos annexos que poden ser útils al lector i que he cregut necessaris posar.



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 2
Anàlisis de situació actual

2.1 CIMNE i l'anterior infraestructura

2.1.1 Sobre CIMNE

La seu principal del CIMNE s'ubica a l'edifici C1 del Campus Nord de la Universitat Politècnica de Catalunya, Barcelona.

Aquest centre fou fundat per quatre professors de la UPC l'any 1987 essent un d'ells l'actual director, Eugenio Oñate. CIMNE és un dels primers centres autònoms d'investigació i desenvolupament creat a partir d'un decret de la Generalitat de Catalunya, DOGC 841-20.5.1987 en col·laboració amb la UNESCO. Té personalitat jurídica com a consorci entre la Generalitat de Catalunya i la UPC. La facturació per projectes de I+D+i a l'any 2009-10 fou de 14.240.257€.

CIMNE organitza un gran nombre d'esdeveniments i activitats dedicades a l'ensenyament i a la divulgació del coneixement en format de cursos, seminaris, conferències i publicacions. També es responsable de moltes activitats relacionades amb la recerca i desenvolupament i ha participat en un gran nombre de projectes de transferència de tecnologia en cooperació amb més de 150 empreses i organitzacions de diferents països. El resultat d'aquesta transferència de tecnologies, i també de coneixement en un context internacional està amplament detallat a la seva pàgina web *http://www.cimne.com* [1], i avarca la organització de més de 470 cursos i seminaris, 100 conferencies nacionals i internacionals, la publicació de més de 125 llibres, 15 programes de software educacional, 186 monografies, 2 revistes periòdiques, més de 1000 publicacions científiques i tècniques, la participació en més de 1400 projectes I+D, i un llarg etcètera.

La organització del centre compta amb la seu central que es troba a Barcelona, a l'edifici C1 del Campus Nord de la UPC i per altra banda té seus secundàries a Castelldefels, Terrassa, Madrid, Eivissa i altres internacionals a Washington DC (USA), Santa Fe (Argentina), Singapore i Pekin (Xina). Organitza també el que s'anomenen Aules CIMNE, que són espais de col·laboració en temes docents i de I+D creats conjuntament per CIMNE i un o més grups universitaris. Les aules s'ubiquen principalment per tota espanya i per Amèrica llatina i en total n'existeixen 23.

La seva plantilla es divideix a grans trets en personal de recerca, desenvolupament i innovació, i en personal de serveis generals. En el primer camp hi ha totes aquelles persones dedicades a la I+D teòrica, sectorial i transversal, i està format per gent de països de tot el món formant un total, a data 14-03-2011, de 208 persones sense comptar amb beques, visitants i personal no fix.

Per altra banda el personal de serveis, al qual pertany l'autor d'aquest projecte, compta amb diversos departaments compresos en equipaments, formació i difusió, projectes, administració i finances i amb aproximadament 100 persones.

El seu treball està enfocat, però no exclusivament, a l'aplicació de la tècnica dels mètodes numèrics en enginyeria en el camp d'estructures. Els mètodes numèrics són formes de resoldre equacions amb derivades parcials. Per resumir-ho, qualsevol problema del món físic es pot representar mitjançant una abstracció matemàtica d'aquest problema: com aguanta el pes una cadira, com ha de ser una llauna de refrescos, quina forma ha de tenir un pont per resistir les vibracions del pas de vehicles, com ha de ser dissenyada una presa per aguantar la pressió de l'aigua, etc. Totes aquestes preguntes es poden convertir en equacions matemàtiques que posteriorment es resolen mitjançant la tècnica dels mètodes numèrics. Si aquestes equacions representen correctament la realitat, resoldre-les permet realitzar simulacions i anticipar-te a l'èxit i evitar el fracas. Es tracta de reproduir qualsevol fenomen per optimitzar processos, o inclús, calibrar riscos. CIMNE s'encarrega de tot el que envolta als mètodes numèrics en tots els camps als que s'apliquen, desenvolupa software per la simulació i el càlcul, realitza projectes pràctics, teòrics, estudis, etc. Els mètodes numèrics són aplicables a camps com l'enginyeria civil, aeronàutica, enginyeria aeroespacial, medicina, intel·ligència artificial, geomecànica i un llarg etcètera.

2.1.2 L'anterior servei de càlcul de CIMNE

Durant la dècada dels '90 i coincidint amb l'increment de la potència dels ordinadors i de la globalització de Internet, CIMNE va decidir invertir en la tasca d'informatitzar totalment el centre mitjançant l'estructuració i planificació de la xarxa i posterior adquisició d'equips informàtics per els treballadors i investigadors. Durant el temps les necessitats computacionals augmentaren degut a la naturalesa de l'estudi en el camp dels mètodes numèrics i a les simulacions que es realitzaven. Per aquest motiu, l'any 2008 es planificà la creació d'un centre de processament de dades i la organització integral de tots els components que formaven el sistema així com la unificació de la sala de servidors i millores en el manteniment.

Un dels punts més significatius d'aquella activitat va ser la decisió d'adquirir un clúster de càlcul per tal de donar suport als nombrosos investigadors que requerien més potència de càlcul.

Per tal d'adquirir el clúster es va haver d'entrar en un concurs com es fa típicament en l'entorn universitari i amb empreses que reben diners de fons públics.

Les empreses que van entrar en el concurs van ser:

- Ànima BCI/SUN – *http://www.animabci.com*
- BULL - *http://www.bull.es/*
- Unitronics/SUN - *http://www.unitronics.es/*
- Dell - *http://www.dell.es*

Les diferents propostes es van avaluar i les puntuacions quedaren reflectides a la Taula 1.

Criteris Avaluació Exp 01/2008	Referència		Puntuació							
	Valor	%	BULL		Dell		Unitronics/SUN		AnimaBCI/SUN	
Valoració Estratègica	**10**	**100%**		**2**		**4,8**		**6,9**		**5,9**
Diversificació	2	20%	2	0,4	2	0,4	4	0,8	7	1,4
Aliances estratègiques	3	30%	2	0,6	3	0,9	7	2,1	5	1,5
Qualitat proveïdor	5	50%	2	1	7	3,5	8	4	6	3
Valoració Econòmica	**40**	**100%**	5,0018644364	**20,0075**	10,00003636	**40,0001**	6,2969509091	**25,1878**	5	**20**
€		**63.800**	63.796		53.223		61.056		63.800	
Valoració Tècnica	**50**	**100%**		**20**		**39**		**32**		**33**
Valoració del risc	15	30%	2	3	8	12	8	12	6	9
Proposta Tècnica	15	30%	4	6	8	12	6	9	8	12
Referencies	10	20%	4	4	8	8	6	6	5	5
Millores	10	20%	7	7	7	7	5	5	7	7
Total Puntuació	**100**			**42,0075**		**83,8001**		**64,0878**		**58,9**

Taula 1. Concurs per adquisició d'una solució HPC – 2008

Com es pot veure el guanyador va ser Dell i efectivament va ser l'escollit per realitzar el projecte. Es pot trobar més informació d'aquest procés a l'Annex 1 Documents interns a la taula amb la valoració realitzada [doc1].

La solució proposada per Dell va ser la d'adquirir un chassis amb 11 servidors de tipus blade, un switch extern de 24 ports amb capacitats iSCSI, un switch Infiniband integrat en el chassis i connectat a tots els nodes i finalment una cabina de discs SAN iSCSI amb 5 discs de 750Gb.

Amb el pressupost també s'inclogué la instal·lació de la solució HPC i la gestió del projecte.

La factura es pot trobar a l'Annex 1 Documents interns al desglossat de preus de Dell, [doc2].

En quant al software, Dell va subcontractar l'empresa *Linalco Consulting S.L.* per realitzar el desplegament del sistema operatiu a tots els nodes del clúster, la instal·lació d'unes cues de treball i la posada en marxa de tot el sistema.

La solució inclogué la compra de les 13 llicències necessàries del sistema operatiu RedHat Enterprise WS 4 update 5 (2007-05-01, kernel 2.6.9-55), versió específica per computació d'alt rendiment, juntament amb la instal·lació del paquet *O.S.C.A.R.*, un software que inclou tot el necessari per automatitzar la instal·lació d'un clúster de càlcul. Es va fer l'esquema de la xarxa i es van instal·lar també algunes biblioteques de càlcul.

El nom determinat pel clúster en general i pel node màster va ser “Acuario”, i per la resta de nodes fou “pez001 a pez012”.

A partir d'aquell moment i com a única documentació, es deixà un sol informe d'instal·lació de 14 pàgines descrivint el procés realitzat i un parell de proves bàsiques fetes. L'informe d'instal·lació es pot trobar a l'Annex 1 Documents interns a [doc3].

Addicionalment al clúster de càlcul Acuario i com a suport als investigadors que no tenien necessitats de realitzar càlculs en paral·lel es va optar per la instal·lació de dos servidors de la casa Sun Microsystems. Aquests servidors es mencionaran més endavant amb el nom de XFire (veure Figura 1) i Vega, ambdós amb sistemes GNU/Linux.

És important també fer referència a un tercer servidor anomenat “Totalsamba”. Aquest proporcionava emmagatzemament extern i compartit a XFire, Vega i Acuario, però a mitjans de 2010 els seus discs van fallar i es va retirar del CPD sense proporcionar-se cap solució alternativa.

Figura 1: Fotografia d'un servidor Sun Fire x4600

Per obtenir més informació del sistema muntat es poden consultar les presentacions del Café CIMNE que es va realitzar el 2008 en relació als equipaments de càlcul adquirits [doc4] i a l'espai compartit amb TotalSamba [doc5].

2.2 Utilització del servei

2.2.1 Tipus de càlcul realitzats

Els càlculs que realitzen els investigadors del CIMNE són principalment:

- Càlculs en sèrie: 44% dels investigadors
- Càlculs en paral·lel usant OpenMP: 51% dels investigadors
- Càlculs en paral·lel usant MPI: 4% dels investigadors
- Càlculs mixtos amb OpenMP+MPI: 1% dels investigadors

Les mesures són aproximades ja que varia el percentatge en funció del projecte que realitzin. Han estat obtingudes agafant una mostra del total dels investigadors.

Com podem veure la major part dels usuaris treballa fent servir càlculs en sèrie i altres en paral·lel fent servir OpenMP. Això significa que NO s'aprofita tot el potencial del clúster actual que disposa de la xarxa Infiniband d'altes prestacions.

2.2.2 Programari utilitzat

Els investigadors empren diversos frameworks de treball en paral·lel i sobretot orientats als mètodes numèrics. El més important és Kratos Multi-physics, de lliure distribució i desenvolupat per CIMNE.

En la majoria dels casos els resultats que obtenen són visualitzables pel programa GiD, software post i pre-processador de simulacions numèriques, també desenvolupat per l'empresa.

Com a llenguatges de programació fan servir:

- C++
- Fortran
- Python
- Tcl/Tk

Com a biblioteques hi ha molta varietat, però les més importants són:

- Boost – Conjunt de bibliotèques C++ generals optimitzades
- ParMETIS – Bibliotèca basade en MPI que implementa algorismes per particionar grafs, malles, i per calcular reduccions de matrius.
- SuperLU – Bibliotèca de propòsit general per solucionar sistemes d'equacions no lineals i no simètrics en màquiens de càlcul d'alt rendiment.
- Trilinos – Projecte basat en desenvolupar algorismes i crear tecnologíes amb un framework orientat a objectes per solucionar problemes de multi-physics i cientifics.
- Blas - Basic Linear Algebra Subprograms, rutines que proporcionen construccions de blocs bàsics per operar sobre vectors i matrius.
- Lapack – Escrit en Fortran 90 proporciona rotines per solucionar sistemes d'equacions lineals de forma simultània i altres problemes relacionats.
- Papi – Biblioteca que permet capturar events del processador i per tant necessari per els programes de debug.

I com a compiladors i software addicional:

- GCC
- CLang
- Intel® Compiler
- Matlab
- GiD
- CMake

2.2.3 Usuaris i gràfiques d'utilització

Hi ha un total de 37 usuaris donats d'alta al clúster Acuario i uns altres 21 als altres dos servidors de càlcul XFire i Vega. Els logs dels sistemes XFire i Vega demostren com només hi ha dos usuaris que fan servir aquests servidors, mentre que Acuario registra una activitat quasi completa amb tots els usuaris utilitzant-lo de tant en quant.

Posem a continuació algunes gràfiques representatives de la càrrega de treball que ha sofrit el clúster en els darrers mesos.

Cluster Acuario Load last year – La Figura 2 mostra la càrrega mitjana global. La mesura s'agafa prenent el nombre total de processos que s'estan executant en tots els nodes. Com veiem el moment que més processos hi ha en execució és el de principis de març. Durant el mes d'Agost es pot veure una caiguda del treball, clarament indicat pel moment en que UPC talla l'electricitat al Campus Nord de la Universitat. A final del mes d'Octubre es veu també una davallada de la utilització que és deguda a la implementació del nou sistema descrit en aquest document.

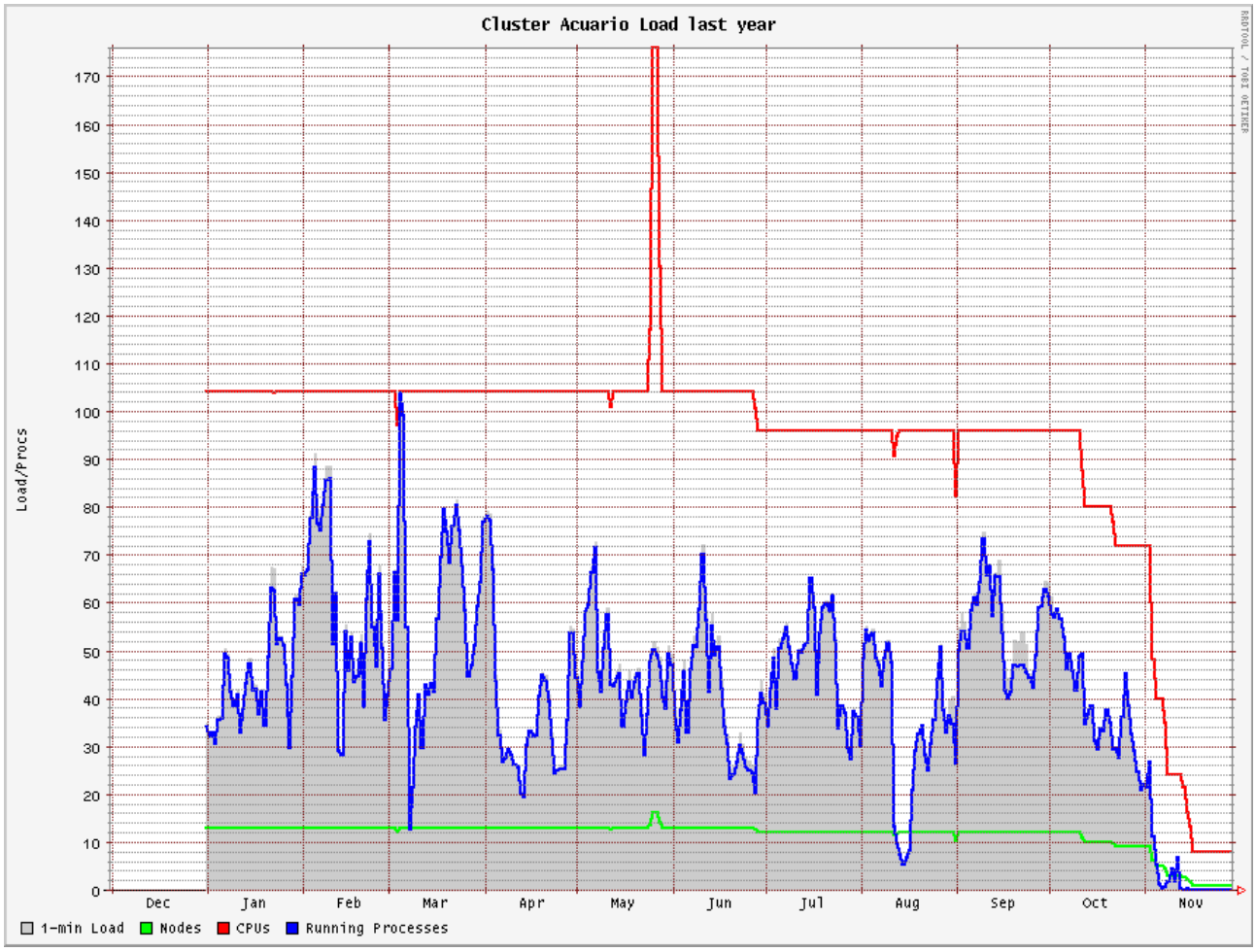


Figura 2: Càrrega de treball del Cluster Acuario (antic) per l'any 2011.

*Cluster Acuario CPU last yea*r – La Figura 3 mostra la utilització de CPU en global pel clúster. La part blava representa la mesura típica de sistemes Linux de CPU d'usuari, mentre que la vermella la de sistema.

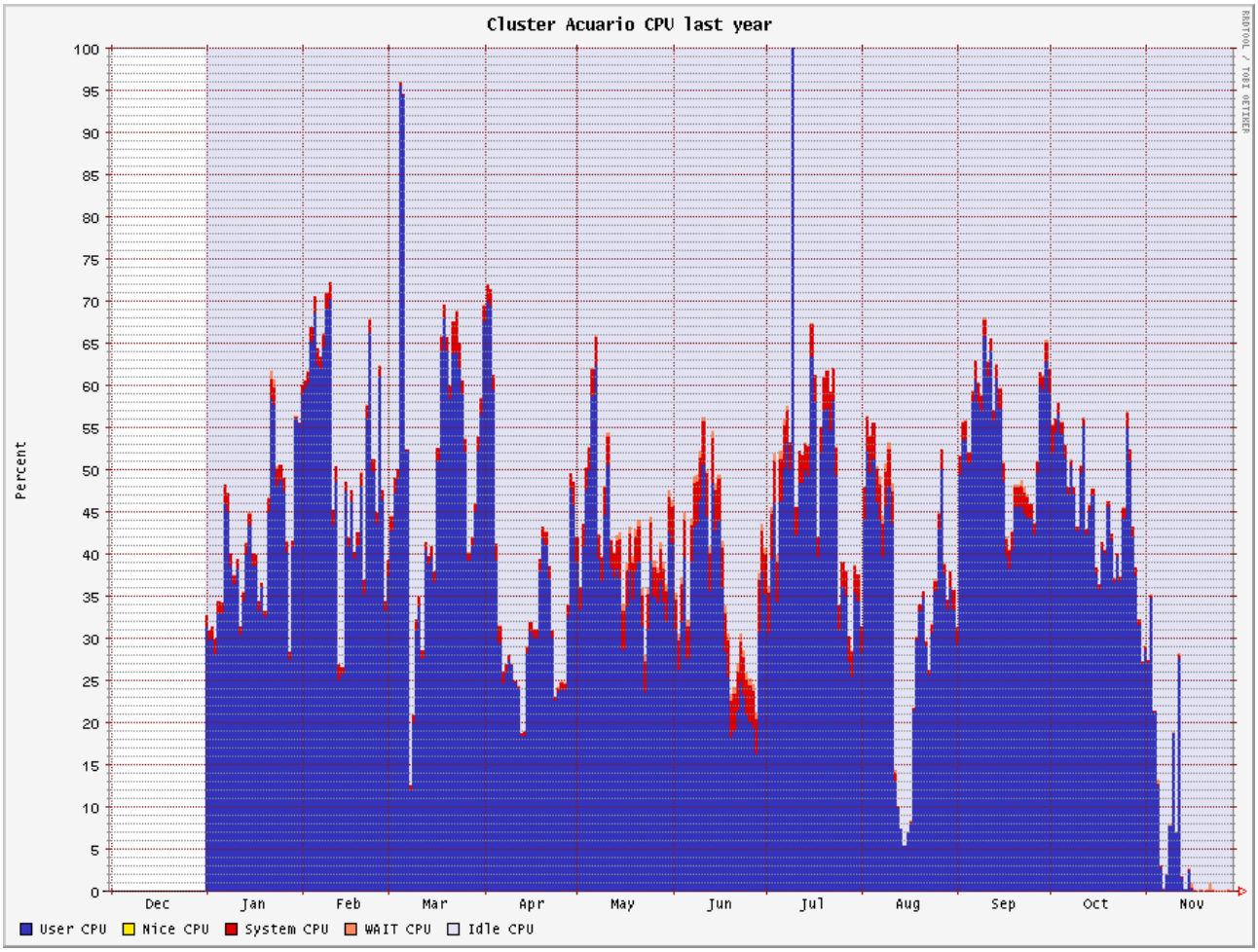


Figura 3: Càrrega de CPU del Cluster Acuario (antic) per l'any 2011.

Cluster Acuario Memory last year – El total de la memòria era de 11 Intel® Nodes * 16Gb + 1 Intel® * 32Gb + 2 AMD * 32 Gb = 272Gb. A la Figura 4 podem veure com a principis de març hi ha una davallada de la memòria consumida. Això significa, juntament amb la gràfica anterior que ens mostra una càrrega del sistema molt elevada, que hi va haver algun problema amb el cluster o que va quedar totalment lliure de treballs.

Finalment veiem que d'aquests 272GB no se'n sol utilitzar, no obstant això la partició “swap” es fa servir, pel que vol dir que alguns nodes si que en ocasions es sobrepassa el llindar de RAM d'algun node.

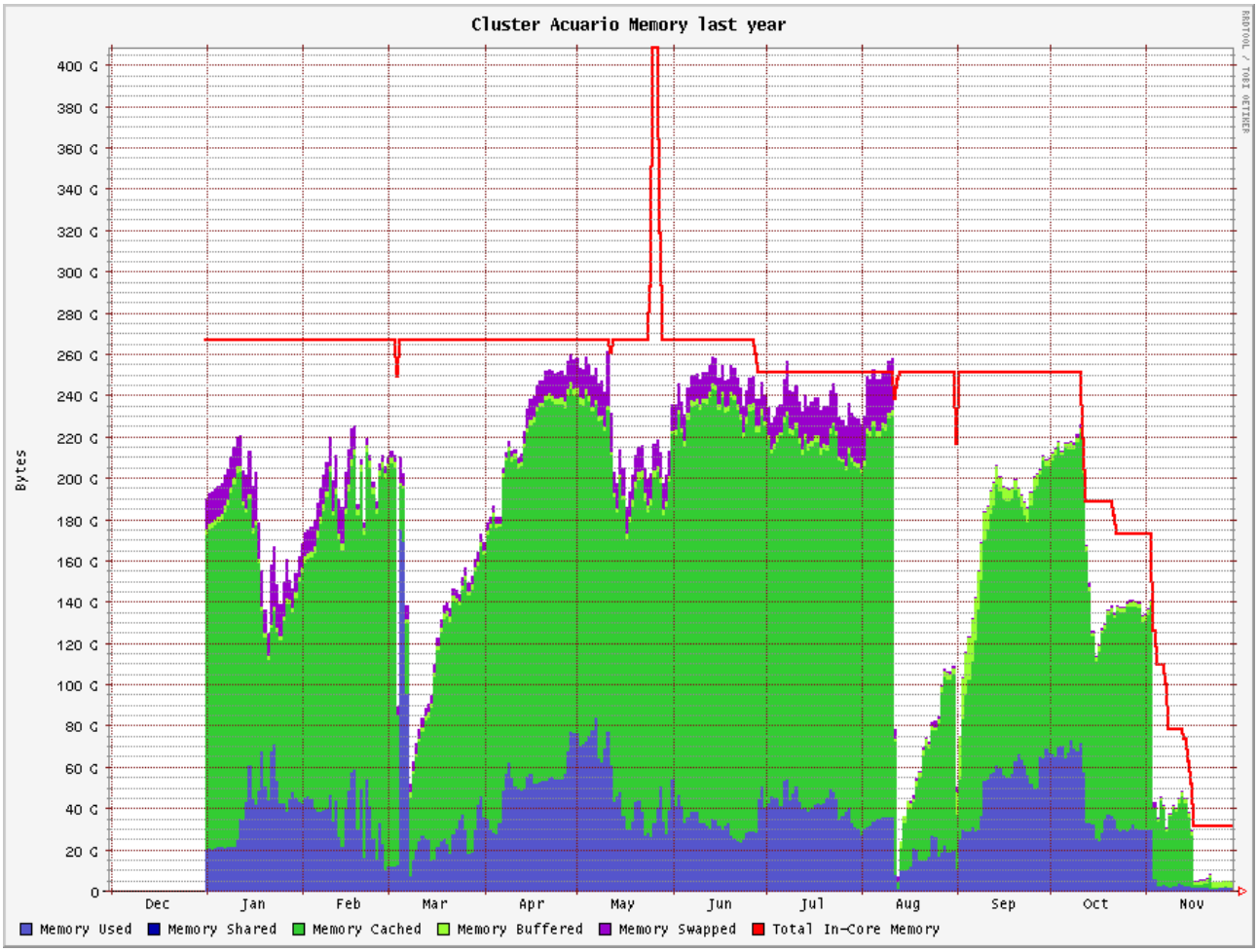


Figura 4: Memòria utilitzada pel Cluster Acuario (antic) per l'any 2011.

Cluster Acuario Network last year – A la Figura 5 veiem que no s'utilitza la xarxa de forma molt intensa. Això és degut a que no s'aprofita correctament la xarxa Infiniband i no es programa en codi paral·lel utilitzant el model MPI.

En els mesos de Maig a Juliol, l'increment de comunicacions és degut al treball realitzat que es descriu en aquest document.

El tràfic dels mesos de Setembre és degut a la migració d'informació d'usuaris a sistemes de backup, per tal de preparar-se per la nova implementació.

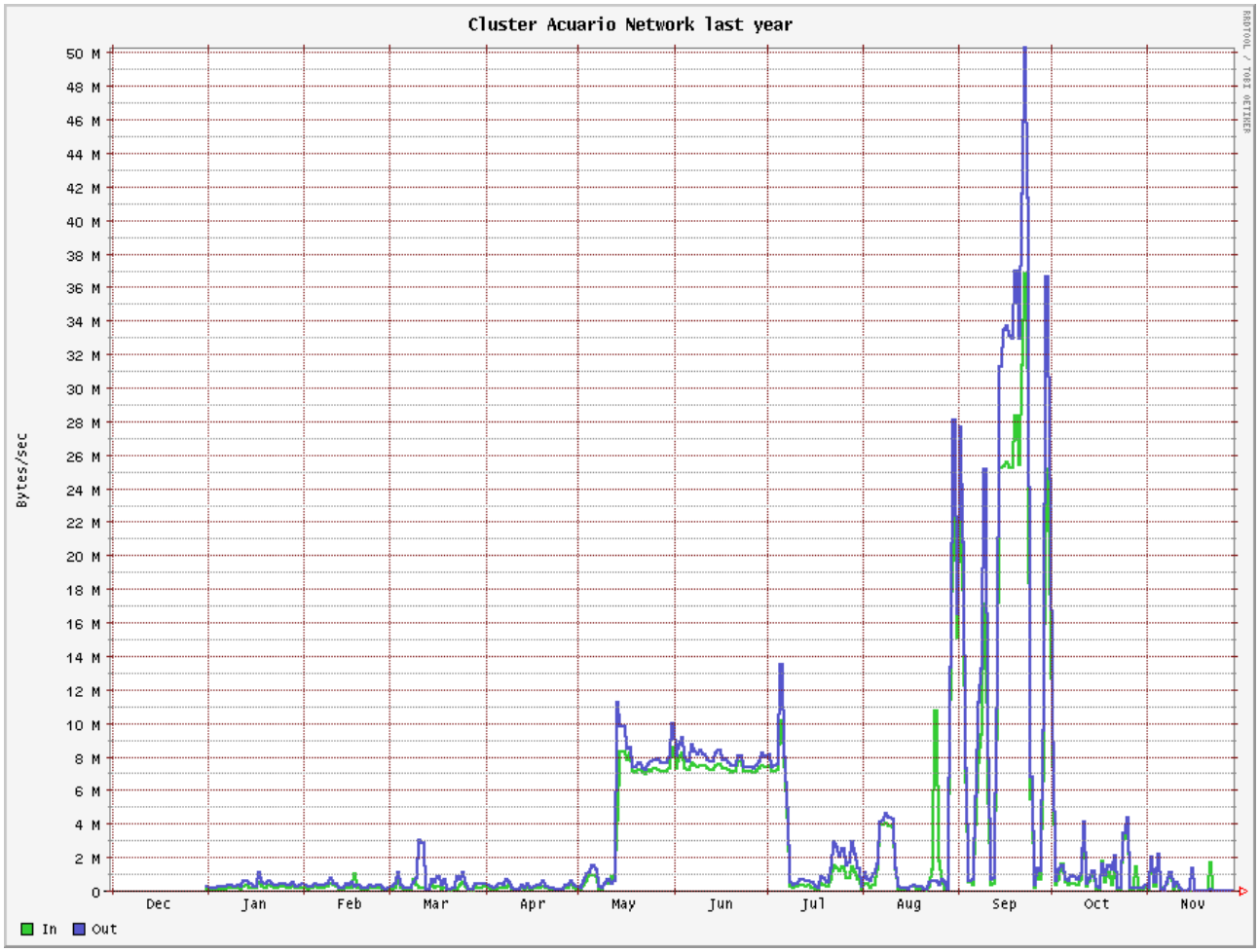


Figura 5: Càrrega de la xarxa del Cluster Acuario (antic) per l'any 2011.

2.3 Problemes detectats

A continuació analitzem els problemes que CIMNE sofria a la infraestructura de càlcul instal·lada en 2008.

Els punts principals són els següents:

2.3.1 Obsolescència de versions

2.3.2 Gestió dels treballs

2.3.3 Deficiències d'instal·lació

2.3.4 Deficiències de documentació

2.3.5 Falta de coneixement de la infraestructura

2.3.6 Ampliació de la infraestructura

2.3.1 Obsolescència de versions

La instal·lació realitzada per *Linalco S.L.* del clúster Acuario comprengué la instal·lació del paquet *O.S.C.A.R.*

Aquest paquet desenvolupat per l'Open Cluster Group té anys de reconeixement i va ser amplament utilitzat en entorns de clúster de tipus beowulf. Facilitava molt la instal·lació i configuració de tots els components del clúster, com la configuració de xarxa, els paràmetres necessaris per el correcte funcionament de les cues, el gestor de recursos, el desplegament de imatges de sistemes operatius, la sincronització d'usuaris i grups entre nodes, etc.

No obstant, el treball i l'esforç de l'Open Cluster Group ha estat minvant durant els últims anys i ha perdut popularitat. La última versió publicada a data d'avui és la d'*O.S.C.A.R.* 6.1.1, de 8 de Febrer de 2011 completant més d'un any i mig sense cap versió nova. Aquesta versió a més només tenia suport per uns pocs sistemes basats en RedHat Enterprise 5, essent la versió actual la 6 update 2.

Amb tot això la instal·lació realitzada fou la d'*O.S.C.A.R.* 5.0, de 12 de Novembre de 2006, i durant l'Abril de 2011, moment en que es començà aquest projecte, el programari que aquesta gestionava ja estava totalment obsolet.

A banda d'aquests problemes la documentació d'*O.S.C.A.R* no es troba actualitzada i està poc mantinguda amb els problemes evidents que això comporta.

Per altra banda, el sistema operatiu instal·lat no fou realment el que s'havia comprat (RedHat EL 4.5 ws). Va ser instal·lada la versió 5, nom clau “Tikanga”, del que no he pogut obtenir informació d'on es va obtenir, tampoc el responsable actual del departament sap com es va instal·lar aquesta versió mentre es tenia llicència per una versió anterior. Aquesta versió del sistema operatiu juntament amb les versions dels seus paquets de software quedaren obsoletes temps abans de l'inici d'aquest projecte.

A mode d'exemple el compilador GCC instal·lat es trobava en la versió 4.1.2-14, compilada a data de 26-06-2007. El compilador de Intel® es trobava en la versió 10.1 de 12-03-2008.

Els investigadors necessitaven les millores de les versions del compiladors i biblioteques que han anat sortint des de llavors, la millores fetes en els nous kernels en quant a paral·lelisme, i també la possibilitat de fer servir nou programari de treball amb les seves noves funcionalitats.

2.3.2 Gestió dels treballs

O.S.C.A.R incloïa el gestor de recursos *TORQUE* i el planificador de treballs MAUI, ambdós programaris mantinguts actualment per el grup *Cluster Resorces* i *Adaptative Computing*.

TORQUE és un gestor de recursos també amplament reconegut, i MAUI és la versió gratuïta, de codi obert i lliure distribució del popular MOAB, també mantingut per *Adaptative Computing*.

No obstant això, encara que aquests programes són reconeguts i possiblement molt potents, es van acabar per no utilitzar a l'entorn del clúster. El motiu ha arribat a ser desconegut però tot apunta a que per algun motiu se li van donar permisos als usuaris o no es va restringir mai la possibilitat de que aquests accedissin individualment als nodes i poguessin executar sense cap tipus de control els programes que ells volguessin.

Al no existir un control sobre les cues de treball, s'ha preguntat a diversos usuaris la forma en que executaven els seus processos i la resposta ha estat unànime en la majoria dels casos. De les respostes dels usuaris hem obtingut un flux de treball que il·lustrem a la Figura 6.

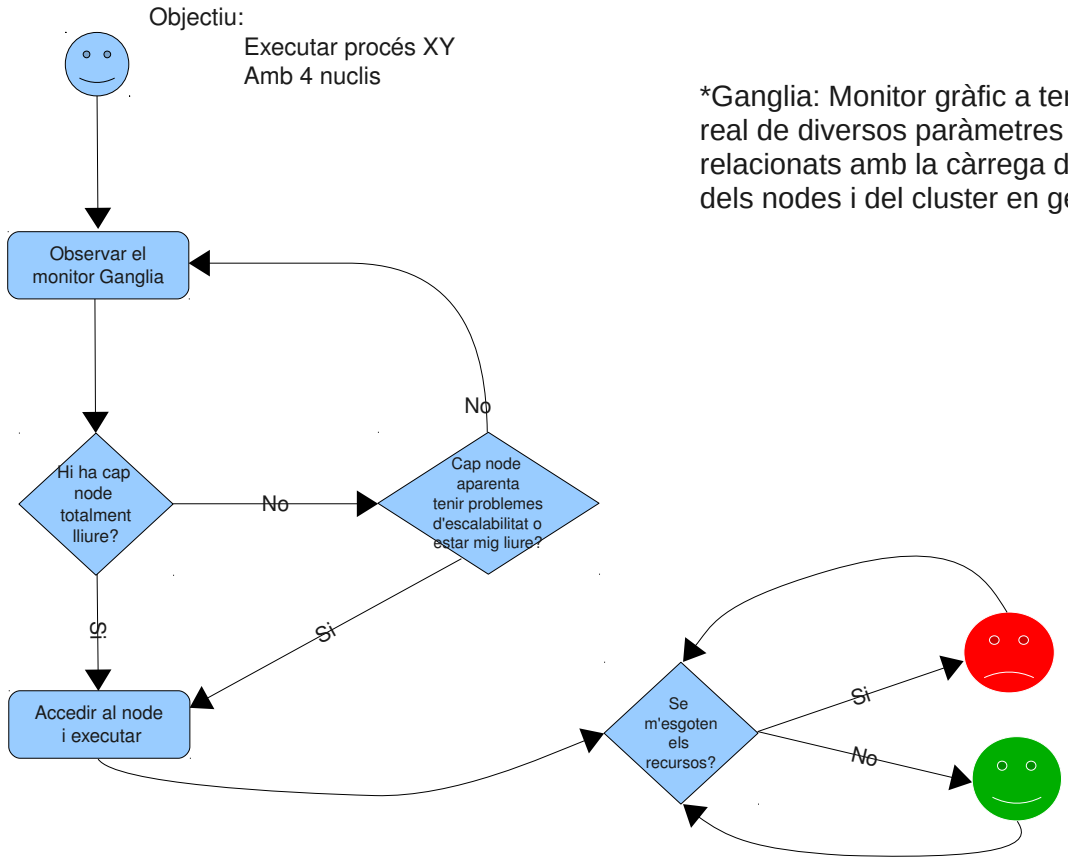


Figura 6: Flux de treball típic d'un usuari en la antiga infraestructura

- En el cas en que l'usuari hagi arribat a la cara verda del final, haurà de preocupar-se constantment durant la seva execució de que cap altre usuari accedeixi al node que està utilitzant i que li suposi un esgotament dels recursos del node.
- En el cas en que l'usuari hagi arribat a la cara vermella final, haurà de prendre mesures per tal de que el procés no se li cancel·li. Per fer-ho haurà de mirar quins usuaris estan treballant en el node en que ha llançat el procés i contactar amb ells per tal de demanar-lis que cancel·lin el seu treball.

Habitualment cap d'aquests usuaris voldrà cancel·lar el seu treball i per tant el node esdevindrà col·lapsat. A més, és possible que algun usuari es salti el pas d'observar el monitor Ganglia o faixi incorrectament la justificació subjectiva de *"Cap node aparenta tenir problemes d'escalabilitat o estar mig lliure?"* i accedeixi a calcular a un node en que no convé gens que ho faixi. Llavors se li esgotaran els recursos a ell i també als usuaris que estiguin calculant en aquell node.

Com a dada curiosa alguns usuaris comentaven que la forma de com determinar si cap node aparentava estar "mig lliure" era accedir al monitor Ganglia i fixar-se en les gràfiques que presentava aquest (Figura 7). Per exemple, en la gràfica següent veiem com el node "eth_pez005" té una càrrega molt elevada i per tant teòricament l'usuari no es plantejaria entrar-hi a calcular. El "eth_pez001" té una càrrega mitja però bastant uniforme, pel que suposa que els recursos s'estan aprofitant prou bé i no convé utilitzar-lo. En canvi, el node "eth_pez013" o "eth_pez014" tenen unes gràfiques que pugen i baixen molt sovint, cosa que indica que en alguns moments el procés que hi està corrent no està fent res (habitualment degut a males pràctiques de programació/espera de E/S) i en altres moments està fent servir tots els recursos disponibles.

La decisió de l'usuari seria la d'entrar a calcular al node "eth_pez013" o "eth_pez014", nodes que li permetran aprofitar tots els moments de càrrega baixa.

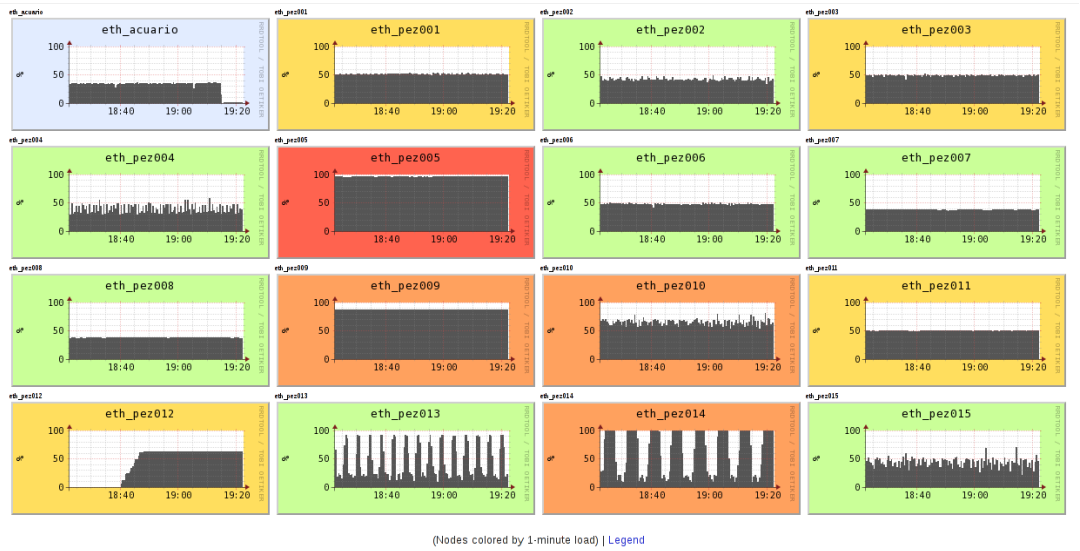


Figura 7: Captura de pantalla del monitor Ganglia

Una altre possible decisió podria ser la d'executar el càlcul en el node màster ("eth_acuario" a la Figura 7). Aquesta decisió seria una de les més perjudicials ja que aquest és el node que controla les comunicacions del clúster, la unitat NFS compartida, la cabina de discs, etc.

Aquests problemes greus de concurrència apareixen sempre que no existeixi un planificador de treballs i un gestor de recursos funcionant i causen un flux de treball molt poc òptim i en ocasions perjudicial entre diferents usuaris. Es trenca tota concurrència segura possible i en moltes ocasions causa que els usuaris s'enfadin i mal utilitzin o deixin d'utilitzar el clúster fent perdre molt de temps i també molts diners a l'empresa.

Aquest motiu va ser el principal motiu i la causa de moltes discussions, problemes amb usuaris, frustracions al no tenir la disponibilitat necessària, i especialment, es formà una visió interna molt negativa de la gestió del departament de sistemes i de l'entorn de computació del que es disposava.

2.3.3 Deficiències d'instal·lació

2.3.3.1 Seguretat del sistema

La instal·lació realitzada per *Linalco* no va preveure que els usuaris es saltarien el sistema de cues i el gestor de recursos. Per aquest motiu no va portar a terme cap tipus de mesura de seguretat en quant a l'accés als nodes i va decidir deixar obert l'accés per SSH a tot usuari a tots els nodes. A més es van deixar diversos usuaris no pertanyents al departament de Sistemes amb el nivell d'administrador mentre que únicament Sistemes és el que s'hauria d'haver encarregat del manteniment i la gestió del clúster. Aquests administradors amb el temps van anar realitzant modificacions sense comunicar-ho als altres ni al departament de sistemes i es va acabar per tenir un descontrol de tot el que succeïa a Acuario i als peixos.

A més no es va establir cap tipus de tallafoc i es va obrir a l'exterior el node màster per tal que els investigadors poguessin accedir lliurement per SSH. No es va implantar cap sistema de seguretat per bloquejar els atacants per força bruta i a més no hi havia política de contrasenyes. El switch que proporcionava connectivitat al clúster no estava programat i per tant no hi havia VLANs, cosa que feia que qualsevol node pogués accedir a qualsevol dispositiu connectat al switch (p.ex. interfície de gestió de la cabina de discs).

2.3.3.2 Còpies de seguretat

A banda de tot això el sistema de còpies de seguretat que s'establí fou el de mantenir en un directori protegit una còpia periòdica del directori /etc i altres que poguessin contenir dades de configuracions. Aquestes còpies no s'exportaven ni es van exportar mai a l'exterior i per tant en cas de qualsevol fallada, per molt que es disposés d'un sistema raid 5 al node màster, podia fer perdre tot el sistema.

2.3.3.3 Desplegament d'imatges

Un altre problema va ser el del sistema de desplegament d'imatges a nodes. No es va documentar com fer-lo servir i a més no es va proporcionar cap sistema per actualitzar les imatges dels nodes. Certament s'utilitzava un sistema documentat per el grup *O.S.C.A.R* però la posterior instal·lació de software especific, com les biblioteques compilades específicament per alguns investigadors i la instal·lació de noves versions de programari no es van incloure en aquestes imatges. El sistema quedà obsolet en poc temps. Això significava que si un dels nodes hagués fallat, no s'hauria recuperat fàcilment. La causa de fallada no era tant improbable ja que cada node disposava de només un disc físic.

2.3.3.4 Comunicacions

Per altre banda, les comunicacions quedaren amb una configuració tocada manualment per *Linalco* que feu dubtar de la seva fiabilitat. Al reiniciar el clúster, en ocasions la xarxa Infiniband no s'aixecava i provocava que tampoc es muntessin les unitats de xarxa ja que aquestes anaven muntades per NFS a sobre d'Infiniband, no a sobre de la xarxa ethernet. Al succeir un fet així s'havia de reiniciar altre vegada tot el clúster i tornar a provar que funcionés, sense rebre cap explicació ni entendre perquè això succeïa. Per altra banda i relacionat amb aquest problema, i per algun motiu que per ara no s'ha arribat a descobrir, el node que controlava la Infiniband no era sempre el mateix. Això vol dir que després d'un re-inici del sistema possiblement el node màster ja no era l'arbitre i s'havia d'esperar a que tots els nodes engeguessin per tal de tenir l'arbitre activat i posteriorment reiniciar el màster i muntar les unitats de xarxa a tots els nodes.

En un moment determinat es va fer una prova amb el programa John The Ripper per tal de determinar quines de les contrasenyes emmagatzemades a /etc/shadow d'Acuario eren més vulnerables.

JohnTheRipper va ser capaç, amb menys de 4 minuts, de desxifrar amb èxit un total de 5 contrasenyes d'usuari.

2.3.3.5 Sincronització de comptes

Un exemple més de problemes d'instal·lació van ser els scripts que es van instal·lar al crontab referents al sistema d'alta i baixa d'usuaris i grups. Cada 15 minuts s'executava un script que sincronitzava els fitxers /etc/shadow, /etc/passwd i /etc/groups a tots els altres nodes. Si coincidia el moment en que es donava d'alta un usuari o grup o es canviava la contrasenya d'algun usuari, amb el moment d'execució d'un d'aquests scripts, o si algun node estava caigut en el moment de l'execució, podien ocórrer situacions indesitjables. La probabilitat de tenir aquestes situacions era del 6,7% (en algunes ocasions es donaren aquests problemes).

2.3.3.6 Sistema d'actualitzacions

Com a últim punt problemàtic de la instal·lació de *Linalco* trobem que el sistema es va modificar en alguns punts i feu impossible l'actualització estàndard d'O.S.C.A.R. Proves realitzades demostren com fallaven les actualitzacions del sistema i com yum reportava molts de problemes amb conflictes entre fonts. No hi havia definit cap procediment per actualitzar els nodes. També hem explicat anteriorment el problema de la Obsolescència de versions i les imatges de sistema que no es van poder actualitzar.

2.3.4 Deficiències de documentació

L'informe d'instal·lació de 14 pàgines que deixà *Linalco* fou insuficient per la gestió posterior de la infraestructura. A més durant l'estudi d'aquest treball s'han detectat errades molt importants com per exemple l'esquema de xarxa, en que es van confondre totes les adreces IP de la xarxa Infiniband amb les de la xarxa ethernet.

A més no es va deixar cap tipus de pla de contingència, cap tipus de manual o esquema de procediment per realitzar les tasques bàsiques de gestió (encesa, apagat, actualització) ni cap documentació de l'estructura de hardware (excepte l'esquema de xarxa) ni de com utilitzar adequadament els recursos disponibles.

Això va fer que es fessin esforços per part dels usuaris principals per crear una documentació de com s'havia d'utilitzar la infraestructura, i a tal efecte es creà una wiki que pretenia informar de tot aquest procés. També es va realitzar una presentació que ja hem comentat anteriorment i que es donà a un “Café CIMNE” [doc4], [doc5] per Miguel A. Pasenau.

La wiki va ser escrita durant unes poques setmanes i no es va completar. Paral·lelament el departament de sistemes començà a elaborar una documentació de com fer servir les cues, però tampoc s'acabà mai.

Per la part d'administració no va existir mai cap documentació excepte uns apunts a la Wiki interna de Sistemes referents a quines comandes executar per donar d'alta i eliminar usuaris.

2.3.5 Falta de coneixement de la infraestructura

El punt seguent que comentem aquí és el que fa referència a la falta de coneixement de la infraestructura. Està molt relacionat amb l'apartat anterior de Deficiències de documentació però també inclou la situació en que es trobava el departament de sistemes responsable del clúster, i de la situació que això causava als usuaris.

La falta de documentació i la càrrega de treball del departament va fer que durant el temps es perdés el coneixement que s'havia obtingut en el moment de la instal·lació de la infraestructura. Tota documentació va quedar reduïda al que hem comentat a l'apartat anterior i a quatre apunts presos a mà per el responsable del departament.

Tota aquesta no-documentació va portar el tenir una màquina i una infraestructura molt potent a uns nivells de utilització molt per sota del rendiment assolible, amb moltes crítiques i un malestar generalitzat. A més era inadmissible que un departament disposés de tal sistema i no disposés dels recursos necessaris per fer front a totes les demandes dels seus usuaris.

Un exemple d'això és que alguns usuaris ens demanaren instal·lar alguns programes per comparar els rendiments dels seus codis, i fou impossible determinar la forma de fer-ho per tots els motius comentats anteriorment. La impossibilitat d'instal·lació d'aquest software va fer que l'usuari perdés el respecte per aquesta infraestructura i decidís calcular a un altre lloc.

Alguns usuaris que utilitzaven el clúster ho feien per realitzar la seva tesis. Al fer un treball d'investigador necessiten una eina fiable i que funcioni, i no era el cas del clúster de CIMNE que no els hi permetia escalar el codi correctament degut al poc control de concurrència que del que es disposava. Resultava frustrant, i és comprensible, el que un usuari estigués treballant en un treball tant important com és la seva tesis i que al cap d'una setmana de tenir el procés funcionant, el node es penges perquè algú altre havia entrat i havia consumit el triple dels recursos que podia consumir!.

Un altre problema freqüent i que estudiarem més endavant, és que el departament no coneixia bé la màquina que havia comprat. Disposava de 10 nodes amb processadors Intel® E5410 i dos nodes amb processadors AMD Opteron 2345. En el cas dels Intel® els processos habitualment no escalaven igual que en els nodes AMD, i la gent no entenia el perquè.

El departament de Sistemes CIMNE té com un dels seus objectius el de donar suport als usuaris, i en aquell cas no es podia donar una resposta coherent i ben estructurada. Per altra banda freqüentment sorgien qüestions tècniques sobre la màquina de la que es disposava, de la forma de treball, etc. i s'havia de remetre a la documentació inacabada que s'havia realitzat. És per això que en aquest treball estudiarem en profunditat l'arquitectura del sistema i determinarem les seves particularitats.

Finalment, tota la desconeixença del sistema causava intranquil·litat al departament per por a que un dia sorgís algun problema greu. Com hem vist anteriorment el clúster de CIMNE és un punt clau i no es podia permetre la no disponibilitat del sistema.

2.3.6 Ampliació de la infraestructura

Durant el 2011 i degut a les creixents necessitats del centre es van decidir adquirir els tres nodes que faltaven per completar els espais del chassis del clúster. Les gestions es realitzaren durant els primers mesos de l'any i els tres nous nodes van arribar el mes d'Abril.

Després de la valoració de diversos pressupostos es va determinar no acceptar la instal·lació dels nodes ni la renovació del software de la infraestructura.

La configuració final es va ampliar amb 3 nodes PowerEdge M610 (2 x Intel® Xeon® CPU E5645 @ 2.40GHz, 48Gb Ram, total 6+6 cores).

Els preus elevats dels pressupostos feien inviable contractar aquests serveis juntament amb el desconeixement del departament i també dels problemes analitzats anteriorment.

Aquest fou un dels primers motius de que sorgís aquest projecte.

2.4 Requisits funcionals i no funcionals

2.4.1 Requisits no funcionals

Gairebé tots els requisits no funcionals coincideixen amb els típics de qualsevol projecte d'informàtica, no obstant justificarem el perquè de cadascun en ordre de prioritat.

- Operativitat: El sistema ha de funcionar. Ha de poder acomplir les tasques per les que ha estat dissenyat.

- Manteniment: El sistema ha ser mantingut per els administradors del CIMNE i per tant ha d'estar documentat per tal de poder solucionar problemes que apareguin.

- Estabilitat: El sistema ha de ser molt estable ja que el fan servir molts usuaris i de diverses formes, a més de forma intensiva i consumint molts de recursos. Una falta d'estabilitat comprometria la qualitat del servei ofert.

- Concurrència: Al ser un servei de càlcul compartit entre molts d'usuaris ha d'estar garantit que tots puguin treballar al mateix temps.

- Disponibilitat: La disponibilitat és un punt clau per aquesta infraestructura, els treballs solen tenir duracions de dies o setmanes, fins i tot mesos i no es pot permetre cap tall del servei que sigui responsabilitat nostre.

- Seguretat: L'accés al servidor és públic i es troba connectat a diversos servidors de la xarxa interna del CIMNE. Els ordinadors de la xarxa interna també hi poden accedir i per tant es pot comprometre la seguretat del centre. Per altra banda els projectes en que treballen els investigadors i els seus resultats s'emmagatzemen a l'espai del clúster. Finalment permetre l'accés a un usuari no legitim podria comprometre alguns projectes.

  Per tant s'ha d'extremar la seguretat.

  En aquest apartat s'inclou també l'aspecte de realitzar copies de seguretat.

- Rendiment: Es pretén mantenir o millorar el rendiment de l'actual infraestructura aplicant les tècniques que siguin necessàries, per exemple optimitzant paràmetres del sistema operatiu.

- Cost: El CIMNE ha d'obtenir un benefici econòmic clar respecte als pressupostos oferts per altres companyies per realitzar aquesta tasca i a ser possible millorar l'amortització actual.

- Escalabilitat: És un aspecte molt important i que ha de permetre que un cop muntada la infraestructura es pugui ampliar afegint més potència de càlcul, espai d'emmagatzemament, etc. Aquest aspecte també influeix en l'escalabilitat del software, en la possibilitat d'afegir nous programes, actualitzacions, etc.

- Usabilitat: Fins el moment la forma de controlar els treballs era poc intuïtiva i complicada. S'intentarà millorar l'aspecte d'usabilitat oferint les eines necessàries i si cal en mode gràfic. També se millorarà la usabilitat per l'administrador.

- Interoperabilitat: La infraestructura s'ha d'integrar amb els serveis que ofereix actualment el centre de processament de dades i permetre la comunicació i interacció entre components.

- Sostenibilitat: En els temps que corren és necessari fer el major estalvi energètic possible amb l'objectiu de reduir les emissions de CO^2 a l'atmosfera i aconseguir ser el més “verd” possible. Aquest aspecte també influirà en el cost.

2.4.2 Requisits funcionals

Els objectius del present projecte són els de implementar una nova infraestructura completa de càlcul per el CIMNE de forma que compleixi els següents punts:

1. Documentar a nivell d'administrador el hardware de l'actual clúster de càlcul.
2. Comprovar la utilitat dels antics servidors XFire i Vega i decidir-ne el seu futur.
3. Realitzar un esquema de la xarxa.
4. Instal·lar 3 nous nodes comprats l'abril de 2011 per completar la capacitat total del clúster.
5. Renovar el sistema operatiu del clúster de càlcul determinant l'alternativa escollida.
6. Instal·lar els components bàsics al sistema per formar un clúster complet, com pot ser una implementació basada en un paquet O.S.C.A.R o una instal·lació totalment manual.
7. Realitzar una instal·lació el més propera als sistemes actuals, fent que sigui altament personalitzable i al mateix temps senzilla/estàndard i escalable.
8. Configurar el sistema per tal de que els investigadors pugin fer servir tot el software i biblioteques que necessiten.
9. Implementar un sistema de control de comptes d'usuari, un sistema de seguretat, crear polítiques d'ús i restringir permisos dels usuaris. Integrar-ho amb LDAP del CIMNE.
10. Restringir els recursos disponibles al node d'accés i evitar els càlculs computacionals per tal de protegir el sistema.
11. Implementar un sistema de cues únic i funcional, en el que no es pugui evitar l'ús d'aquestes per enviar treballs i es respectin els torns i els treballs dels demés.
12. Valorar les configuracions del sistema de cues per tal de beneficiar al màxim als usuaris.
13. Proporcionar els procediments d'administració i gestió del clúster, a més de permetre el manteniment d'actualitzacions i afegits de paquets de forma senzilla.
14. Disposar d'una documentació d'instal·lació extensa (aquest mateix document), d'administrador concreta, i d'usuari funcional. Formar a l'equip de sistemes i als usuaris sobre l'ús del clúster.
15. Implementar un sistema de còpies de seguretat adequat a les infraestructures del departament.

16. Proporcionar un sistema de monitorització propi de l'estat del clúster. Configurar els components bàsics per la integració en el sistema de monitorització del departament de Sistemes basat en Nagios.
17. Preveure la infraestructura per suportar ampliacions.
18. Realització de proves de funcionament i de rendiments de la xarxa, i estudi de possibles optimitzacions.
19. Realització d'un anàlisis en més profunditat del hardware disponible per tal de facilitar informació a l'usuari com a ajut a la programació del seu codi.
20. Determinar el consum d'energia del clúster i els seus components i identificar les tècniques per reduir-lo, sempre d'acord amb les necessitats dels investigadors.

2.4.3 Requisits addicionals

Addicionalment als objectius anteriors, alguns usuaris ens han fet algunes peticions concretes que els hi agradaria que fossin satisfetes en la nova infraestructura. Es valorarà cada petició i si és viable es satisfarà. Les descrivim a continuació:

- **Distribució de processos serie**

Els nodes Intel® Xeon E5410 (Pez001-010) tenen un bus dual per l'accés a la memòria RAM, cosa que fa que al haver-hi més de 2 processos corrent a la CPU (més de 2 cores actius) que faixin ús intensiu de la memòria, es produïra una baixada notable del rendiment.

En els nodes AMD Opteron 2356 aquest fet no passa degut a que tenen ben dimensionat el bus.

En els nous nodes Intel® E5645 s'haurien de fer proves d'escalabilitat.

La conclusió és que es demana que el gestor de cues sigui suficientment intel·ligent per quan un usuari llança un procés, aquest es col·loqui a un processador amb el menor nombre de cores actius possible.

Això col·lisiona amb l'esquema de cues que es plantejaren en 2008 on es reservaven els dos nodes AMD 2356 per només processos sèrie. El punt de vista de l'investigador que proposa és que si hi ha nodes que no s'estiguin fent servir que no siguin AMD, s'haurien de poder utilitzar mentrestant per processos sèrie.

- **Utilització del node master per executar GiD**

Actualment el node Acuario és un node amb dos processadors Intel® E5410 i amb 32Gb de RAM. Aquest s'utilitza com a node d'accés per SSH. També hi ha muntada una unitat NFS que conté una instal·lació de GiD.

Es demana que es pugui executar el software GiD al node màster tal com es fa ara per tal de visualitzar els resultats dels càlculs sense haver de copiar-los per xarxa als PC's personals.

Altres investigadors noten que no és factible fer servir Vega o XFire encara que també tenguin 32Gb de RAM ja que tenen processadors massa antics.

Finalment qui fa la petició diu que s'hauria de muntar un servidor de discs centralitzat per tal de tenir una única instal·lació de GiD (o d'altre software) en tots els sistemes de càlcul (Vega, XFire, Acuario), i també que s'exportin els resultats dels càlculs realitzats a aquests sistemes d'emmagatzemament.

- **Nom d'Acuario**

Es considera que s'ha de deixar el nom Acuario al sistema per tal de mantenir inalterades les configuracions dels usuaris, així com el nom d'accés remot.

- **Servidor central d'accés a recursos de càlcul**

Proposa la idea de separar el gestor de cues Slurm de l'actual clúster Acuario. Això permetria que els usuaris no haguessin d'utilitzar Acuario per visualitzar resultats de GiD ni per compilar.

- **Compilar a Acuario**

Actualment els usuaris del clúster estan compilant al node màster Acuario. S'hauria de poder discriminar quins usuaris estan compilant a Acuario versus els que estan llançant processos de càlcul. No ha d'estar permès fer servir Acuario per calcular, tot i que si limitem la RAM d'un usuari, no podrà compilar, i si limitem el temps, no podrà executar consoles interactives. Hem de trobar un compromís.

- **Hyper-threading**

Ens demana que desactivem l'hyper-threading ja que per alguns càlculs que realitzen els hi suposa una pèrdua de rendiment. S'ha d'estudiar la opció i l'impacte que pot tenir, a més d'aprofitar per determinar quins altres paràmetres de les BIOS es poden optimitzar.

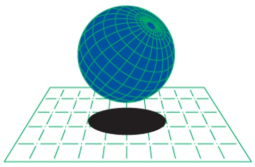
2.5 Conclusions

CIMNE és una empresa que té unes necessitats computacionals elevades. Degut a diversos problemes amb la infraestructura de computació de la que disposen i aprofitant una ampliació de tres nodes nous en el seu clúster de càlcul s'ha determinat que la millor opció per acomplir els requisits del centre i dels seus investigadors, d'entre totes les que se li ofereixen, és la d'assignar a aquest projecte a l'estudiant Felip Moll i a que es realitzi un projecte final de carrera que permeti implementar una solució definitiva.

Es cerca una solució robusta, segura, escalable i eficient que tingui en compte tots els paràmetres característics d'aquests tipus de sistemes. S'ha de realitzar la implementació sense parar el servei actual i per tant s'ha de dur a terme una planificació estricte i amb metodologia. A més es requereixen coneixements tècnics propis d'un enginyer en informàtica superior, especialment enfocats a xarxes, computació i administració de sistemes, i és per això que es designarà aquest projecte com a idoni per un projecte final de carrera.

La solució permetrà a l'empresa disposar d'una infraestructura de computació ajustada a les seves necessitats i integrada amb l'actual centre de processament de dades i els seus sistemes. Es proporcionarà un sistema de cues que permetrà als investigadors enviar treballs al clúster de càlcul de forma ordenada i justa per tots els usuaris. Es proporcionarà també un emmagatzemament de dades controlat i estudiat en termes de rendiment. Es definiran sistemes de comunicació pels usuaris així com la documentació d'usuari i d'administració. S'integraran els sistemes de seguretat i de copies de seguretat del CPD. Es determinarà la millor solució per aprofitar els antics equips de càlcul. S'asseguraran les actualitzacions del software i biblioteques utilitzades pels investigadors. També es realitzaran estudis del hardware i un anàlisis de consum que permetrà a CIMNE tenir el control sobre el cost de la infraestructura i poder determinar el benefici que en treu.

Finalment es pretén reduir també el cost del projecte així com les emissions de CO^2 equivalent a l'atmosfera, aspecte molt important en els temps que corren.



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 3
Gestió del projecte

3.1 Metodologia i eines

El projecte seguirà, encara que no estrictament una metodologia ITIL v3 [3] que n'assegurarà la correcta evolució. Aquesta metodologia determina un conjunt de bones pràctiques per els departaments de les TIC i inclou també les recomanacions pel que han de ser les fases d'un projecte.

La correlació de fases de ITIL v3 amb la documentació d'aquest projecte és mostra a la Taula 2.

Etapa ITIL v3	**Capítols de la memòria**
Estratègia del servei	Anàlisis de situació, Gestió del projecte
Disseny del servei	Investigació, Implementació
Transició del servei	Desplegament
Operació del servei	Eines de suport i documentació, Estudis tècnics
Millora continua del servei	Millora continua i futur

Taula 2: Correlació capítols de la memòria amb fases ITIL v3

Com a eina principal de suport al projecte farem servir el servei de CIMNE Project Manager (basat en Redmine [4], Figura 8) + SVN [5], que ens disposa d'una interfície web amb calendaris, gestor de tasques, diagrames de Gantt, càlcul del temps, wiki interna, etc. I el repositori de codi basat en SVN.

Utilitzarem també un PC personal (Dell Vostro 400) amb Debian GNU/Linux 6.0 i la xarxa del CIMNE.

Disposarem d'altres eines proporcionades per l'empresa com accés a la possibilitat de disposar de màquines virtuals, discs durs i altres materials.

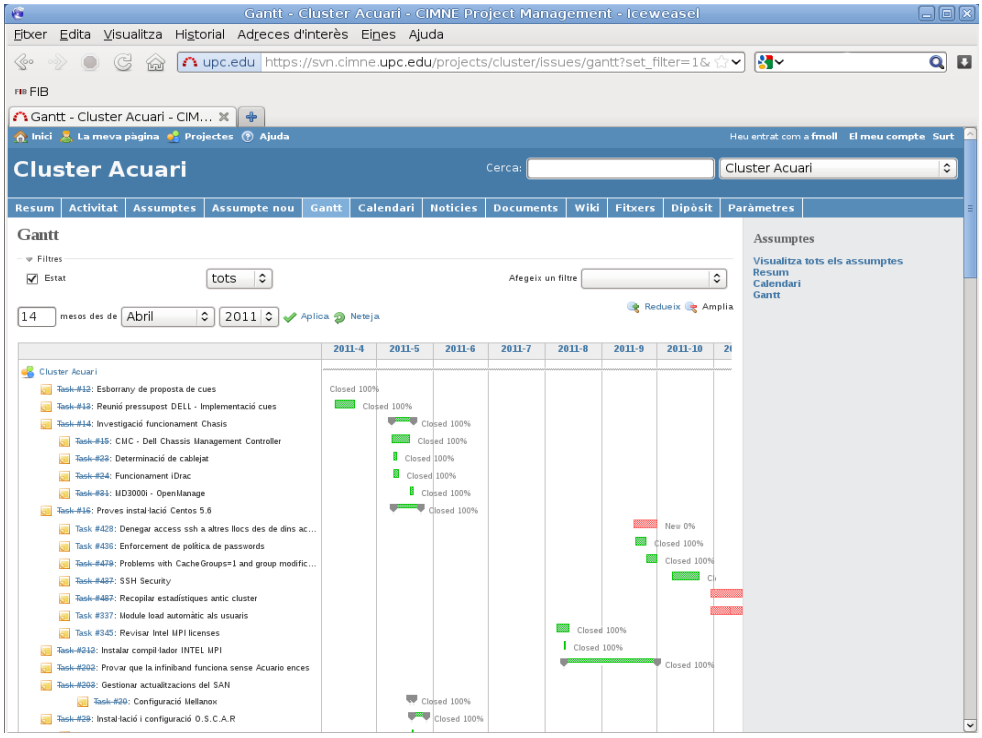


Figura 8: Gestor de projectes Redmine CIMNE

3.2 Planificació temporal inicial

Inici:	1 Abril 2011
Entrada en millora continua:	1 Octubre 2011
Temps de feina previst:	817 hores
Jornada laboral:	4 hores (mitja jornada), amb vacances durant el mes d'Agost
Cost previst:	11.907,81 €

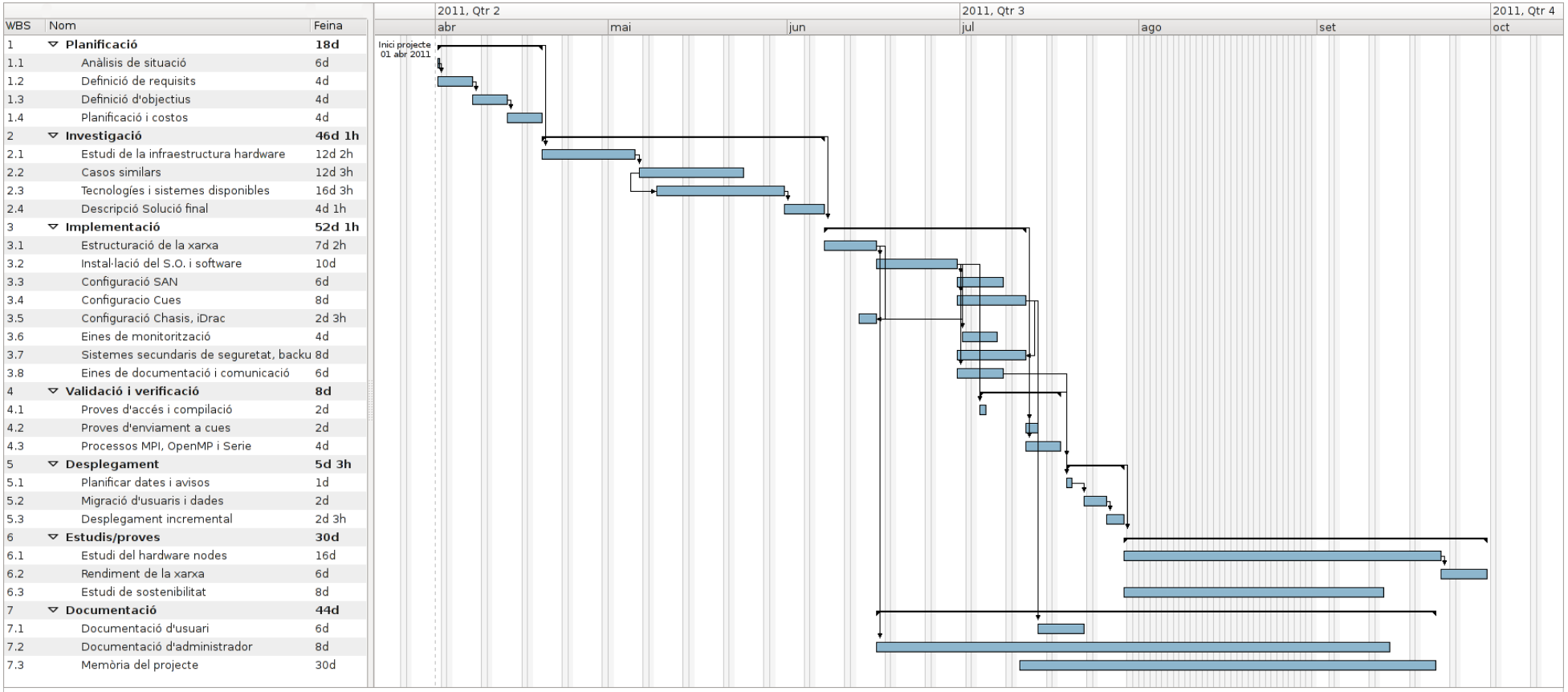
El projecte s'iniciarà l'1 d'Abril de 2011 i s'entrarà en millora continua a principis d'Octubre de 2011 amb un total de 817 hores de treball per part de l'estudiant a jornada reduïda (4h) i un cost estimat de 11.907,81€. La planificació es detalla a continuació.

El comput de temps de totes les etapes suma un total 204 dies laborals de feina constant sobre el projecte. Com que la jornada és reduïda en aquest cas, un dia compta com a 4 hores i el càlcul surt a 817 hores en total repartides de la següent manera entre les fases:

- Planificació: 72h (18 dies)
- Investigació: 185h (47 dies)
- Implementació: 208h (53 dies)
  - Verificació i validació: 32h (8 dies)
- Desplegament: 23h (6 dies)
  - Avaluació, estudis i proves: 120h (30 dies)
- Documentació: 176h (44 dies)

El detall de cada etapa i del temps calculat per cada sub-tasca d'aquestes etapes es troba detallat a la Figura 9 que ve a continuació.

Per fer els càlculs ens hem basat en l'experiència prèvia en alguns projectes ja realitzats i en el temps que ha suposat cada etapa.



WBS	Nom	Feina
1	**Planificació**	**18d**
1.1	Anàlisis de situació	6d
1.2	Definició de requisits	4d
1.3	Definició d'objectius	4d
1.4	Planificació i costos	4d
2	**Investigació**	**46d 1h**
2.1	Estudi de la infraestructura hardware	12d 2h
2.2	Casos similars	12d 3h
2.3	Tecnologíes i sistemes disponibles	16d 3h
2.4	Descripció Solució final	4d 1h
3	**Implementació**	**52d 1h**
3.1	Estructuració de la xarxa	7d 2h
3.2	Instal·lació del S.O. i software	10d
3.3	Configuració SAN	6d
3.4	Configuracio Cues	8d
3.5	Configuració Chasis, iDrac	2d 3h
3.6	Eines de monitorització	4d
3.7	Sistemes secundaris de seguretat, backu	8d
3.8	Eines de documentació i comunicació	6d
4	**Validació i verificació**	**8d**
4.1	Proves d'accés i compilació	2d
4.2	Proves d'enviament a cues	2d
4.3	Processos MPI, OpenMP i Serie	4d
5	**Desplegament**	**5d 3h**
5.1	Planificar dates i avisos	1d
5.2	Migració d'usuaris i dades	2d
5.3	Desplegament incremental	2d 3h
6	**Estudis/proves**	**30d**
6.1	Estudi del hardware nodes	16d
6.2	Rendiment de la xarxa	6d
6.3	Estudi de sostenibilitat	8d
7	**Documentació**	**44d**
7.1	Documentació d'usuari	6d
7.2	Documentació d'administrador	8d
7.3	Memòria del projecte	30d

Figura 9: Planificació temporal i diagrama de gantt del projecte

3.3 Anàlisis de costos inicial

L'anàlisi de costos l'efectuarem sobre la planificació temporal i sobre l'ús dels equipaments necessaris per portar a terme el projecte.

3.3.1 Mà d'obra

El sou del treballador és de 14,5€ bruts la hora, i el temps total de feina previst és de 817 hores, per tant:

Cost del temps de feina: 14,5€ * 817 hores = 11.846,5€

El desglossament d'aquest preu es troba a la Taula 3.

ETAPA	QUANTITAT	COST (€)
Planificació:	72h (18 dies)	1.044
Investigació:	185h (46 dies, 1h)	2.682,5
Implementació:	208h (52 dies, 1h)	3.030,5
Validació i verificació:	32h (8 dies)	464
Desplegament:	23h (5 dies, 3h)	333,5
Estudis / proves:	120h (30 dies)	1740
Documentació:	176h (44 dies)	2.552
TOTAL		**11.846,5 €**

Taula 3: Cost per fase

3.3.2 Amortització dels equipaments

A aquest preu se li ha de sumar el cost d'amortització referent a l'ús dels equipaments bàsics per realitzar el projecte. Consisteixen en un ordinador personal i, fins a l'etapa de desplegament, l'ús de 3 nodes de càlcul per les proves i primeres instal·lacions.

El consum de l'ordinador personal s'estima en un pic màxim de 350W segons les especificacions de la font d'alimentació en un règim estable i continu, tot i que el consum normal no sobrepassa els 150W, mesura que s'agafarà.

El consum dels tres nodes que s'utilitzen durant les primeres etapes fins que **no** s'utilitzen per el càlcul, tenen un consum màxim de 200W segons l'eina iDrac proporcionada per Dell.

S'utilitzarà el preu de 0,170819 €/kWh corresponent a la tarifa 2.1A, consum major de 10kW i no superior a 15kW, obtinguda de Iberdrola [2].

El cost total de l'amortització dels equipaments utilitzats és de 61,31€. El preu queda desglossat a la Taula 4.

MATERIAL	QUANTITAT	COST (€)
Ordinador Personal	817 hores (122,55kWh)	20,93€
3 Nodes durant Investigació+Impl.	98 dies, 2h (236,4kWh)	40,38€
TOTAL		**61,31€**

Taula 4: Amortització dels equipaments

Hem realitzat els càlculs amb la següent formula:

1 W = 1 Joule/seg

1 Hora = 3600 seg

1 kWh és la unitat d'energia equivalent a 1 kW dissipat durant 1 hora, o també 3,6MJoules

P.ex.

Total consumit en J:

150 joule / seg * 817 hores * 3600 seg / 1 hora = 441180000 Joules

Total en kWh:

441180000 Joules * 1 MJoule / $1x10^6$ Joule * 1 kWh / 3,6MJoules = 122,55 kWh

Cost total en €:

122,55 kWh * 0,170819 €/kWh = **20,93€**

*(Una forma més senzilla de fer aquest càlcul és 0,150KW * 817h = 122,55 kWh.)*

Es necessitaran també altres aparells secundaris com un switch de 8 ports 100Base-T i espai d'emmagatzematge temporal, a més d'altres recursos com consumibles per gravar el sistema operatiu, etc. Tots aquests recursos estan ja disponibles al departament i no suposen un increment de cost significatiu respecte el preu del projecte.

3.3.3 Cost final del projecte

Així el total del projecte es determina a la Taula 5.

CONCEPTE	COST (€)
Mà d'obra	11.846,5 €
Amortització dels equipaments	61,31€
TOTAL	**11.907,81€**

Taula 5: Cost final del projecte

3.3.4 Comparativa de pressupostos

A mode informatiu volem mostrar aquí els pressupostos presentats en funció de les ofertes que ens realitzaven empreses externes. Totes aquestes ofertes no contemplaven el total dels objectius de la forma que ho fem en aquest projecte i per aquest motiu es van desestimar.

Pressupost CIMNE (aquest projecte)

Acompliment **de tots els objectius i requisits**	11.846,5 €
Equipaments necessaris	61,31€
Total:	**11.907,81 €**

Pressupost DELL 2008

Instal·lació del sistema operatiu i configuració:	8.000€
Gestió del projecte:	860€
Total:	**8.860€ (sense iva)**

Pressupost DELL 2011

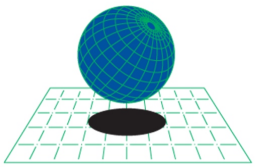
Configuració d'un sistema de cues:	5.148€
8 hores de configuració remota:	832€
Actualització del sistema operatiu:	1.770€
Total :	**7.750€ (sense iva)**

Pressupost SUN

Instal·lació del sistema operatiu i configuració:	~15.000€
Total:	**~15.000€ (sense iva)**

Pressupost UPCNET

Declaren no tenir capacitat per realitzar la tasca.



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 4
Investigació

4.1 Infraestructura hardware

En aquest apartat detallarem quin és la infraestructura de hardware de que disposem. Serà el pas previ per conèixer el sistema i per veure quines possibilitats ens ofereix.

4.1.1 Racks del servei de càlcul i línies d'alimentació

El CPD de CIMNE disposa de diversos racks que allotgen servidors. Un rack no és res més que un armari metàl·lic de mides i ancoratges normalitzats que permet col·locar servidors de qualsevol fabricant de forma fàcil i optimitzada, per exemple permetent passar els cables de forma correcta i ordenada per els seus laterals o facilitant el flux d'aire per el seu interior.

El model de rack del que es disposa a CIMNE és el Dell™ PowerEdge™ 4210 [doc6] Figura 11. Al rack que allotja el clúster de càlcul i la cabina de discs (Rack 3) l'hi arriben quatre línies d'alimentació, dues de 16 amperes i dues de 32 amperes parcialment visibles a la part del terra de la Figura 10.

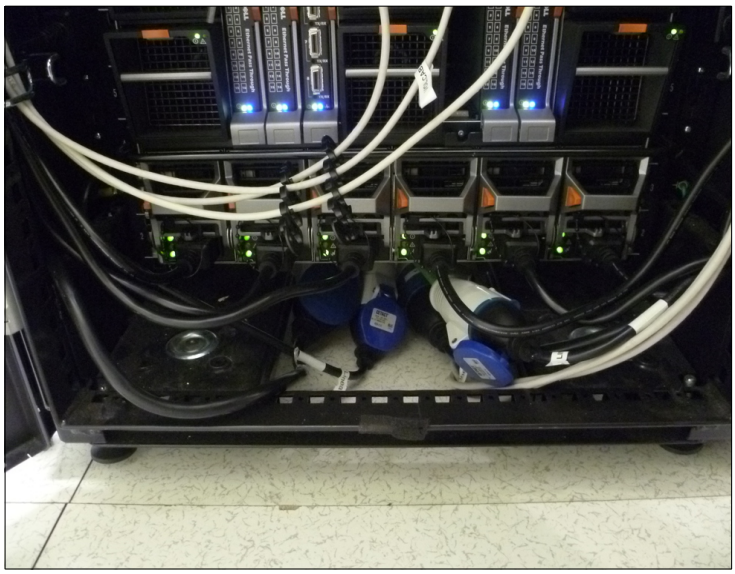


Figura 10: Línies de tensió i fonts d'alimentació del chassis al rack 3.

Figura 11: Rack 3, allotja el clúster i el SAN.

S'han instal·lat als laterals posteriors del rack 3 quatre PDUs de corrent per connectar les PSU dels equipaments. A cada lateral hi ha dos PDUs col·locats en vertical. Els dos superiors de cada lateral disposen de 5 connectors de tipus C13 i 4 de tipus C19. En aquest cas només s'utilitzen els C13 que alimenten el SAN i els switch. Els dos PDUs inferiors disposen cadascun de 4 sortides C19 que alimenten les 6 PSU del chassis de tipus C20.

Els dos PDU superiors s'alimenten de les línies etiquetades com a Línea 11 i Línea 12 (pertanyents a la fase elèctrica 1), mentre que els dos inferiors s'alimenten de les Línea A i Línea B (fase 2).

Resumim la terminologia dels connectors a la Taula 6 i mostrem una fotografia de la ubicació dels PDUs de la línia 11 i 12 a la Figura 12:

Connector	Forma	Parella
C13	Femella	C14
C14	Mascle	C13
C19	Femella	C20
C20	Mascle	C19

Taula 6: Connectors elèctrics

Figura 12: PDUs superiors de tipus C13 i C19. Es veu un cable C14 connectat a un C13.

L'alimentació proveïda a les PDUs es controla des de la caixa de tèrmics instal·lada al CPD. Els diferencials porten la mateixa etiqueta de Línia X que la línia que controlen, Figura 13,

Figura 13: Diferencials de la sala del CPD. En verd els relacionats amb les línies del rack 3.

Per resumir la instal·lació del rack 3, disposem de 4 línies de tensió a 220v, dues de 32Amp i dues de 16Amp que es distribueixen amb 4 PDUs. La intensitat total que arriba al rack és de 32*2+16*2 = 96Amperes, i per tant la potència pic que poden desenvolupar els equips és:

P=220*96 = 21.120W

Remetem al lector al manual d'instal·lació elèctrica per el chassis MD1000e (que veurem a continuació) i que proporciona molt de detall en quant a aquest tema [doc7].

El segon rack que completa la infraestructura de càlcul del centre és el rack 2. Aquest disposa de dues línies de 16 Amperes provinents de la fase 2 i que es despleguen mitjançant PDUs C13 i C14. Aquest rack es comparteix entre dos servidors de càlcul (XFire i Vega) que analitzarem en els següents apartats i alguns servidors no relacionats. Les línies del rack 2 són la Línea1 i Línea2.

Com a mesura de protecció en vers els possibles talls d'electricitat o pujades i baixades de tensió, el CPD de CIMNE compta amb un SAI de la marca SOCOMEC Mastersys [6], que permet aguantar tota la sala en funcionament durant 30 minuts després d'un tall de corrent. Actualment aquest SAI no està habilitat per realitzar l'apagat automàtic dels servidors i només es limita a enviar un e-mail d'avís, fet que s'ha de tenir en compte en el moment de pensar en la possible protecció que se li podrà donar als servidors de la infraestructura de càlcul. L'entrada i la sortida del SAI es corresponen a les 3 fases P1, P2 i P3.

Finalment com a eina de control dels servidors del centre de càlcul es disposa d'un KVM que centralitza tota la gestió quan s'ha d'estar físicament interactuant amb les màquines. Els servidors disposen d'una sortida adequada per la connexió a switchs KVM mitjançant connectors RJ45 i interfície de consola analògica ACI. Permeten una connexió en cascada fent servir switchs amb la interfície ARI. Aquesta tecnologia és de la marca Avocent [8].

Figura 14: KVM amb el tty0 d'Acuario

4.1.2 Cluster

Tots els components del clúster de CIMNE són de la marca Dell™. Es composa d'un chassis i 16 nodes que descriurem a continuació.

4.1.2.1 Chassis

El chassis que es va adquirir el 2008 per CIMNE és tracta d'un Dell™ PowerEdge™ M1000e. Les característiques tècniques concretes i descripcions més extenses d'aquest sistema es poden trobar a la pàgina web de Dell [7] i específicament a l'apartat de suport introduint el *Service Tag* de l'equip. Les descripcions que segueixen han sigut extretes del manual del propietari [10], aquí comentarem les més generals.

Model:	PowerEdge M1000e
Service Tag:	7JT4Q3J
Capacitat de mòduls:	16 de mitja altura, 8 de altura completa, 32 d'un quart d'altura, o combinació de tots. La configuració d'aquest treball és de nodes de mitja altura.

Panell de control frontal

1. Port USB per el ratolí
2. Port USB per el teclat
3. Connector de vídeo VGA
4. Botó d'encesa del sistema
5. Indicador de l'alimentació del sistema

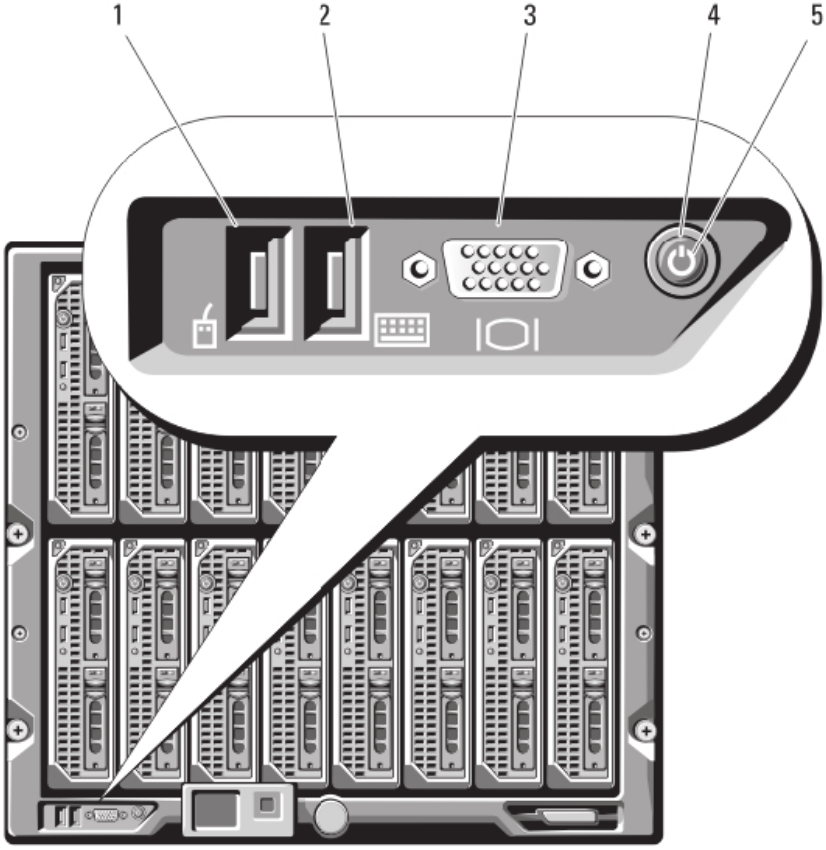


Figura 15: Característiques del panel de control

Mòdul LCD

Ens permet controlar l'estat dels diferents nodes i del chassis d'una forma ràpida i a més és un indicador de problemes ja que visualment es posa de color carabassa si hi ha alguna incidència. També permet configurar el sistema iDRAC del chassis i els seus nodes, monitor de consum, etc.

1. Pantalla LCD
2. Botons de desplaçament
3. Botó de selecció

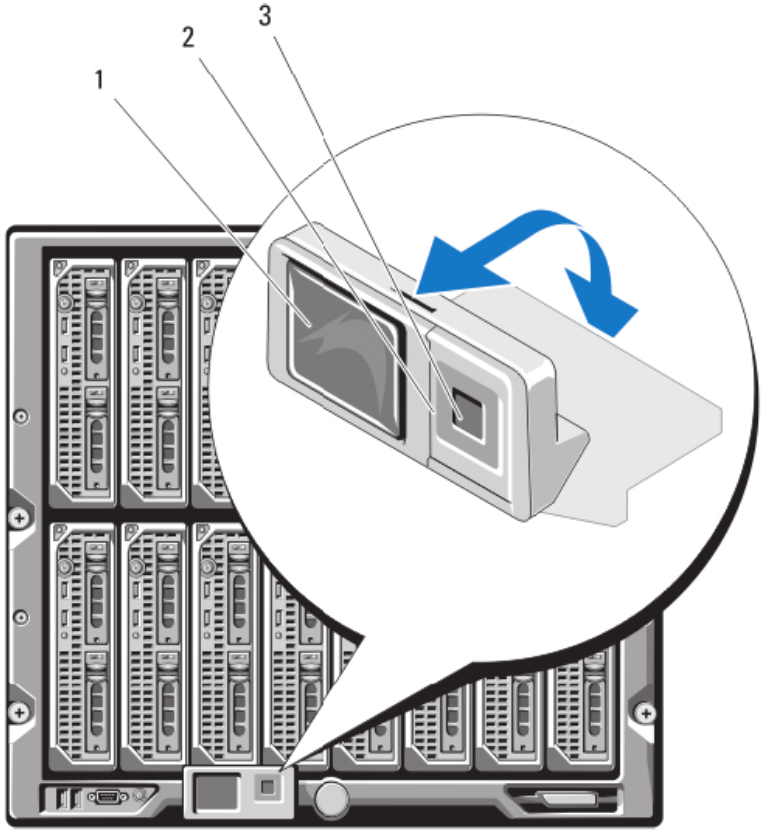


Figura 16: Pantalla LCD

Panell posterior

1. Mòduls de ventilador (9)
2. Mòdul CMC principal (CMC1)
3. Mòduls de E/S (fins a 6)
4. Mòdul iKVM
5. Mòdul CMC secundari (CMC2)
6. Fonts d'alimentació (6 PSU)

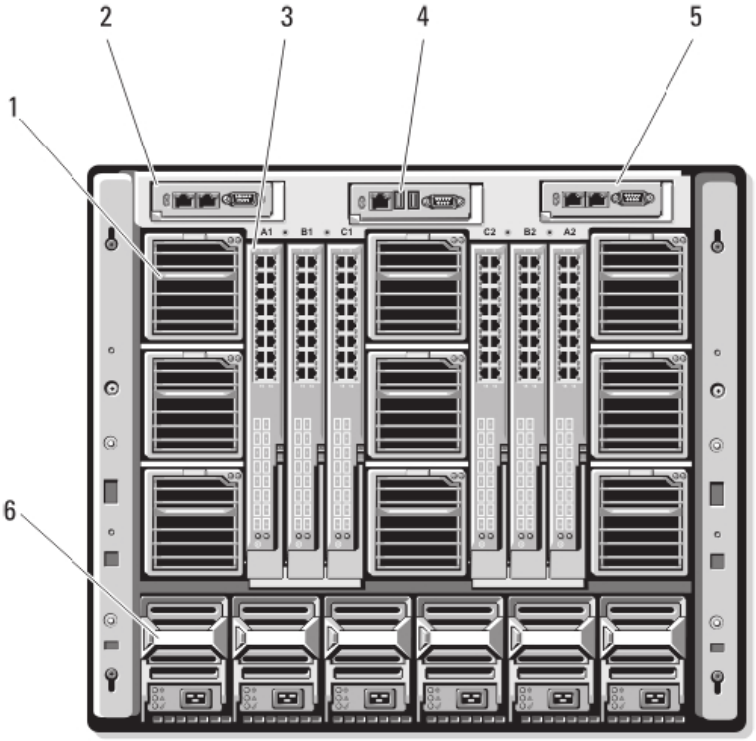
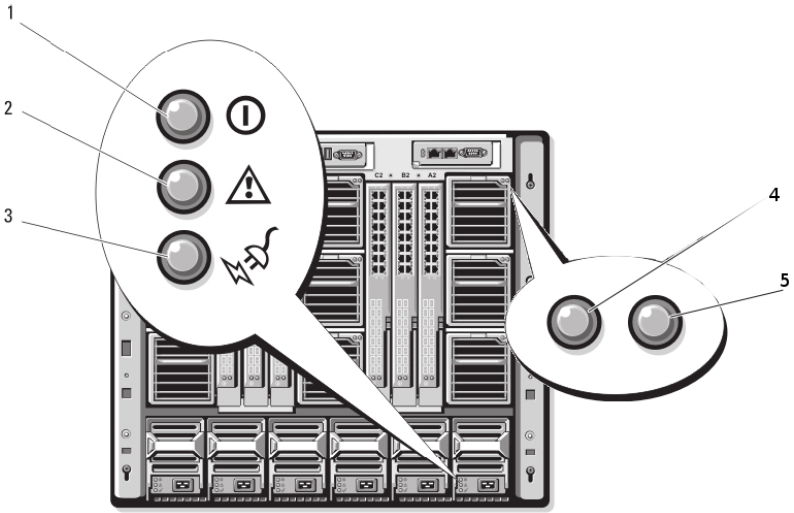


Figura 17: Vista posterior

En el cas que ens ocupa disposem de tot el comentat en els sis punts anteriors excepte els mòduls d'E/S en que els ocupats són els A1, B1, C1, B2 i A2. Per tant C2 queda lliure.

Fonts d'alimentació PSU i ventiladors



1. Indicador de sortida d'alimentació de CC.

2. Indicador de fall a la font d'alimentació.

3. Indicador d'alimentació de CA.

4. Indicador d'alimentació del ventilador

5. Indicador de fall del ventilador

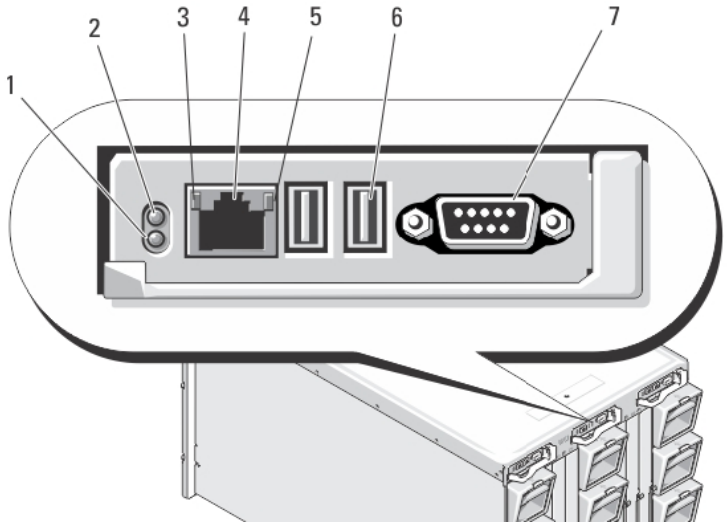
Figura 18: Indicadors de les PSU i ventiladors

Mòdul iKVM

El mòdul iKVM marca Avocent [8] permet connectar un monitor, teclat i ratolí al chassis de forma alternativa al connector VGA del panell frontal. A més mitjançant la interfície de gestió iDRAC que comentarem més endavant permet connectar cada mòdul (node) pel protocol del KVM amb el port RJ-45, fet que ens permetrà controlar els nodes físicament a la pantalla del KVM i no haver de tenir un teclat i un ratolí específics per el clúster.

Nota: iKVM es diferència de KVM en la “i” que indica que és un KVM que utilitza l'iDRAC de Dell.

Més informació a la web de Dell [9].

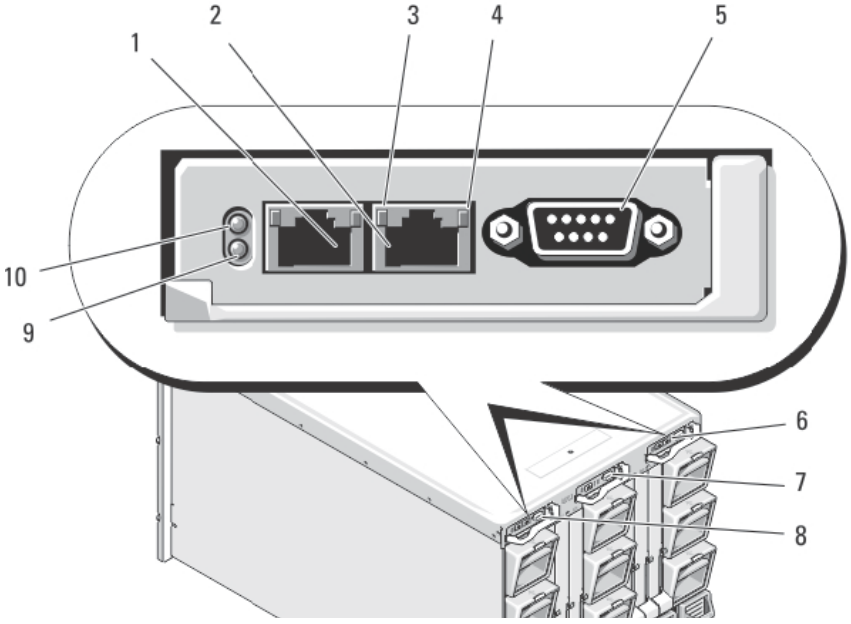


1. LED d'estat / identificació
2. LED alimentació
3. LED d'enllaç
4. Port RJ-45 ACI (només connexions en cascada)
5. LED d'activitat
6. Connectors USB pel teclat i el ratolí
7. Connector de vídeo

Figura 19: Mòdul iKVM

Mòduls CMC – Dell Chassis Management Controller

Aquest mòdul proporciona una interfície web per la gestió del chassis i dels seus nodes mitjançant el protocol iDRAC. En cas de que el CMC1 falli, automàticament s'activa el CMC2 que passa a ser el principal. Si falla el CMC2 tornarà a ser el CMC1 el principal. S'indica l'estat d'error a la pantalla frontal del chassis, i amb un led al propi mòdul CMC.



1. Connector Ethernet 1Gb
2. Connexió Ethernet Stack, per connectar altres CMCs d'altres chassis

3 i 4. LED's d'enllaç i activitat

5. Connector DB-9 sèrie per la configuració local
6. CMC2 secundari, per redundància
7. Mòdul iKVM
8. CMC1 principal
9. LED blau d'estat/identificació
10. LED d'alimentació

Figura 20: Mòdul CMC1 i CMC2

Mòduls d'Entrada/Sortida

Es disposa de tres capes de xarxa diferents d'entrada sortida, aquestes són la Fabric A, Fabric B i Fabric C. Cadascuna de les ranures d'E/S de la part posterior del chassis A1, B1, C1, i A2, B2, C2, admeten un tipus de capa de xarxa. Així els A1 i A2 admeten els Fabric A, els B1 i B2 els Fabric B i els C1 i C2 els Fabric C.

La xarxa Fabric A només admet mòduls de tipus Ethernet, mentre que Fabric B i C admeten mòduls de tipus Ethernet, Infiniband o Fibra òptica.

La xarxa Fabric A, només admet mòduls d'E/S de Ethernet. Cada servidor integrat al chassis, és a dir, cada blade, disposa d'una controladora ethernet Gigabit de xarxa Fabric A integrada que s'enllaça amb l'A1. Així tot servidor al compartiment o *slot 'i'* del chassis disposarà d'una connexió ethernet 1Gb que sortirà pel port *'i'* del mòdul A1.

La xarxa de Fabric B, és una xarxa de 1 a 40Gbps que admet les ranures B1 i B2. Els mòduls a B1 i B2 poden ser de tipus Ethernet 1 o 10Gb, Infiniband DDR/QDR i Fibra òptica de 4 o 8Gbps. Per tal de que un node disposi d'una connexió al mòdul de Fabric B, s'haurà d'instal·lar una targeta controladora adequada ("mezzanine card") dins el node.

La xarxa de Fabric C, és igual que la xarxa de Fabric B però que admet les ranures C1 i C2 en comptes de B1 i B2. De la mateixa forma, perquè un node tingui connectivitat a aquesta xarxa necessita una controladora dedicada.

A la Figura 21 veiem un exemple d'un node a l'slot n=1 amb dues targetes controladores de dos ports cadascuna, Mezz_1 connectada a la xarxa Fabric C, i una Mezz_2 a la Fabric B juntament amb la correspondència de cada targeta i mòdul.

· La targeta de xarxa ethernet integrada es connecta al port *n* del mòdul d'E/S A1, i al port *n* del mòdul d'E/S A2. Al sistema operatiu es veuran com a dues targetes diferents.

· La targeta controladora B, es connecta al port *n* del mòdul d'E/S B1 i al port *n* del mòdul d'E/S B2. Al sistema operatiu es veuran com a dues targetes diferents.

· La targeta controladora C, es connecta al port *n* del mòdul d'E/S C1 i al port *n* del mòdul d'E/S C2.

Atenció, si les targetes controladores dels nodes (Mezzanine cards) no tinguessin dos ports només hi hauria per exemple una connexió del node *n* al port *n* de B1 i no una a B1 i a B2.

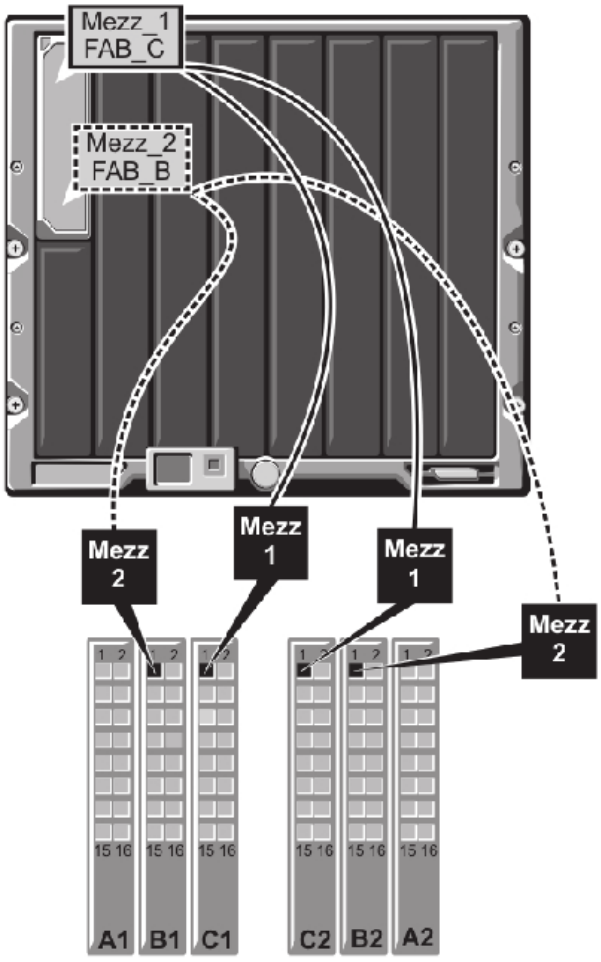


Figura 21: Exemple d'assignació de ports d'un blade de mitja altura

En el cas que ens ocupa tenim establert el següent:

A1, B1, A2, B2	Dell Ethernet Passthrough
C1	Switch Infiniband Cisco M SFS7000E DDR 4x [11]
C2	Buit

Dell Ethernet Passthrough

No és res més que un mòdul que permet obtenir connectivitat exterior de les targetes de xarxa dels nodes. Això fa que necessitem un switch extern per poder establir connectivitat entre ells.

Disposa de 16 ports RJ45 i un ample de banda per port de fins a 1Gbps, no és un mòdul gestionable i es instal·lable en calent.

Switch Infiniband Cisco M SFS7000E DDR 4x

El modul C1 allotja un switch Infiniband de Cisco que disposa de 16 ports 4x interns i connectats als nodes, d'aquesta forma hi ha un port Infiniband per node. Per la part frontal en surten 8 ports més per si es volen connectar dispositius externs.

Per tenir connectivitat amb aquest mòdul de Fabric C, els nodes han de dispossar d'una targeta controladora ("mezzanine card") instal·lada al socket de Fabric C adequat. Habitualment cada controladora d'aquest tipus disposa de dos ports, un que es connecta al mòdul C1 i un que es connecta la mòdul C2. En el cas que ens ocupa, aquestes targetes només estan connectades al C1 ja que el mòdul C2 és buit.

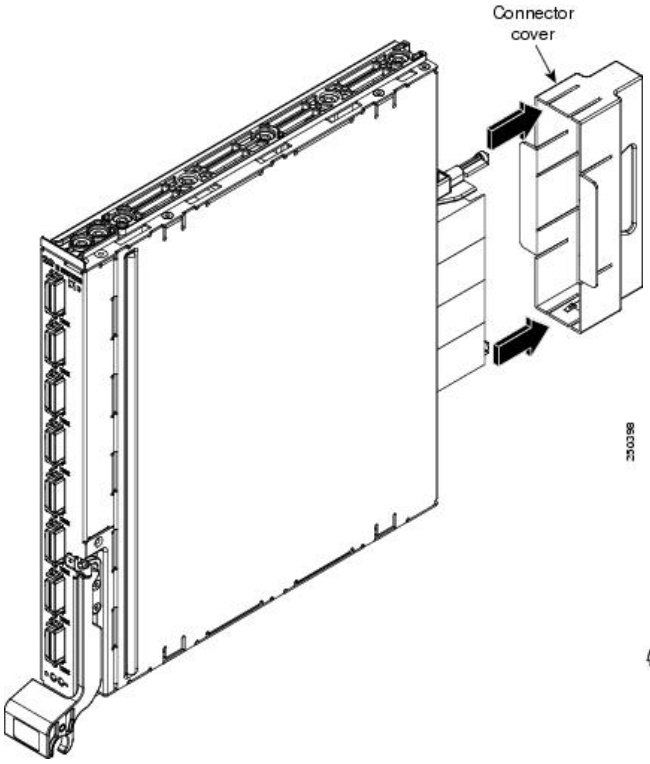


Figura 22: Preparant per instal lar el switch Infiniband

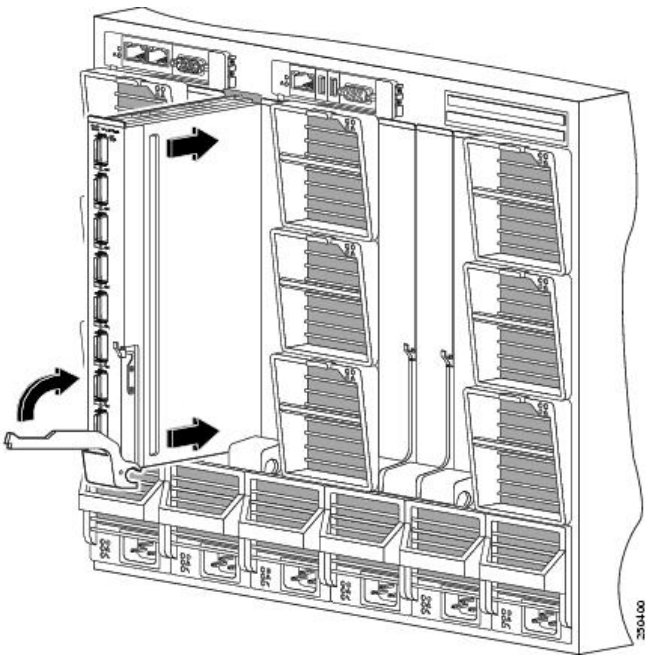


Figura 23: Inserció del switch al chassis

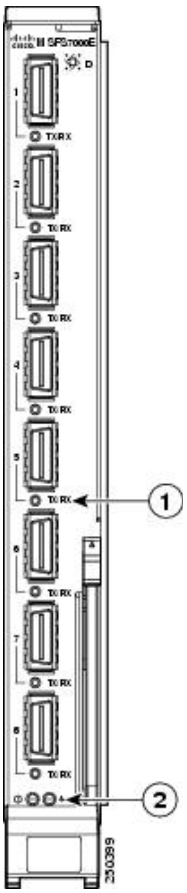


Figura 24: Part posterior del switchInfiniband.

- ***1.*** *Indicador d'activitat,*
- ***2.*** *Indicador d'estat del switch*

A la Figura 24 veiem la part posterior del switch Infiniband i dels 8 ports externs 4x DDR de que disposa.

El switch és gestionable mitjançant eines específiques de CISCO instal·lades en un host connectat al switch [28].

Estudiarem les característiques de l'Infiniband a l'apartat 4.1.4 respectiu a les Comunicacions.

4.1.2.2 Nodes

L'opció que es va escollir en el moment de la compra d'aquest clúster va ser la de nodes de mitja altura de tipus blade.

Al moment de l'inici del projecte es disposava dels següents nodes en funcionament:

- **Node màster** [16]**:**

  1 x Dell™ PowerEdge™ M600 (Intel® Xeon® CPU E5410 @ 2.33GHz, 32Gb RAM)

- **Nodes E5410** [16]**:**

  9 x Dell™ PowerEdge™ M600 (Intel® Xeon® CPU E5410 @ 2.33GHz, 16Gb RAM)

- **Nodes AMD2356** [17]**:**

  2 x Dell™ PowerEdge™ M605 (Quad-Core AMD Opteron™ 2356, 16Gb RAM)

D'altra banda, els tres nous nodes que es van comprar el 2010 i pendents d'instal·lar eren:

- **Nodes E5645** [12]**:**

  3 x Dell™ PowerEdge™ M610 (Intel® Xeon® CPU E5645 @ 2.40GHz, 48Gb RAM)

A continuació descrivim els components de la sèrie PowerEdge™ M6xx que ens ocupa.

Descripció física i integració al chassis

Els nodes són blades de mitja altura (18.9x5x48.6 cm, 6Kg) i son inseribles/extraibles en calent tal com es mostra a la Figura 25. A la Figura 26 podem veure els connectors posteriors que connecten el node al chassis. Són connectors propietaris de Dell.

Figura 25: Node parcialment extret del seu slot

Figura 26: Connector posterior d'un node

Placa Base

La placa base suporta l'arquitectura de memòria compartida i distribuïda entre els dos sockets de CPU. Cada CPU disposa de 6 bancs de memòria (4 en els M600 i M605) que depenent del node poden anar des de 64 fins a 192Gb en total. Disposa també d'una controladora SAS per els discs durs, un dispositiu de descàrrega de CPU per tràfic iSCSI, els slots del Fabric B i C (el Fabric A amb dos ports està integrat a la placa), i una ranura SD per inserir un hypervisor com pot ser VMware ESXi i no requerir disc dur. Mostrem un resum de les parts a la Figura 27.

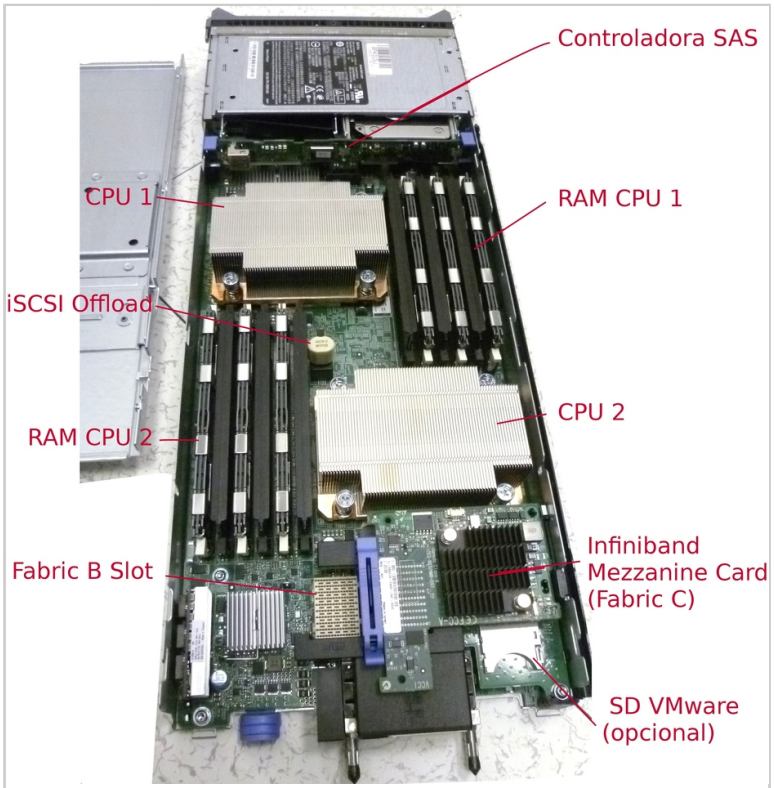


Figura 27: Parts del blade

Discs durs

Tots els nodes disposen d'una controladora SAS CERC6i (SAS only) de Dell i un disc dur Dell Enterprise SAS 6Gbps de 146Gb, 2,5" i 10K RPM, excepte pels nodes E5645 que els discs són de 15K RPM.

El node màster disposa de dos discs durs i la controladora SAS6ir [15] que té capacitats d'efectuar RAID 0 o 1 per hardware, Figura 28.

Els discs són extraïbles en calent tot i que evidentment si només hi ha un disc i el sistema operatiu està funcionant això no serà factible, veure la Figura 29.

Figura 28: Controladora SAS amb 1 disc connectat

Figura 29: Disc parcialment extret del node

Targetes de xarxa

Cada node disposa de dues targetes Fabric A model Broadcom® NetXtreme II BCM5708S (BCM5709S per els M610) 1000Base-SX PCI-X 64-bit 133Mhz.

Al Fabric C de cada node hi ha instal·lat una controladora Infiniband Mellanox ConnectX de dos ports. Als nodes M610 es tracta del model Mellanox Technologies MT26418 [ConnectX VPI PCIe 2.0 5GT/s - IB DDR / 10GigE], mentre que als altres nodes és Mellanox Technologies MT25418 [ConnectX VPI PCIe 2.0 2.5GT/s - IB DDR / 10GigE].

Al Fabric B no hi ha instal·lada cap controladora en cap node excepte en el node principal (slot 1) i en el node 2 (slot 3). En aquests nodes disposem d'una controladora Gigabit Ethernet de 2 ports que s'enllaça als mòduls B1 i B2 als ports corresponents a l'slot de cada node. Figura 30.

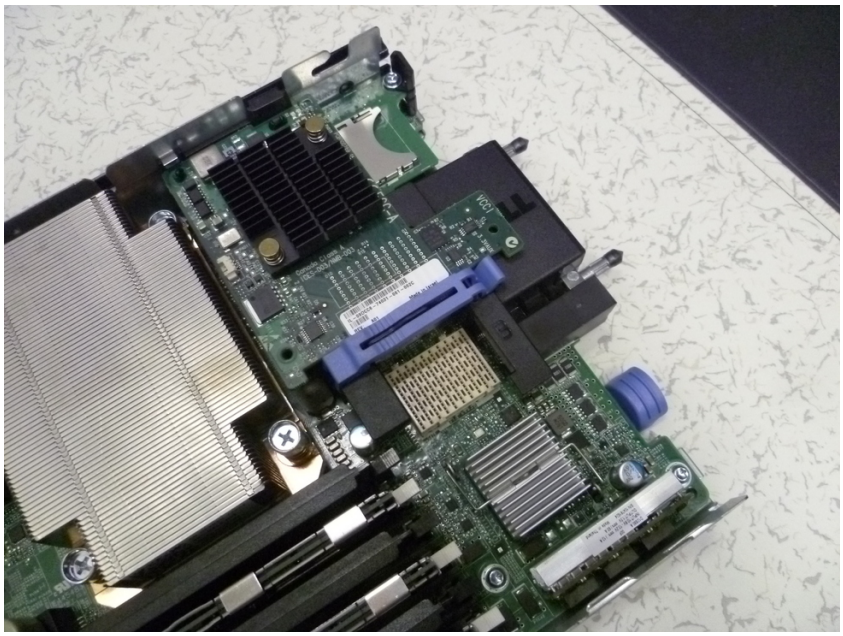


Figura 30: Fabric C amb una controladora Infiniband. Disipador de la controladora Fabric A. Fabric B lliure.

Els nodes M610 disposen també d'un dispositiu descarregador de la CPU per transferències de tipus iSCSI. S'anomena iSCSI Offload Engine (iSOE) i és un petit dispositiu amb connector RJ-45 que connectat a un port de la placa base realitza les operacions de xarxa iSCSI evitant que sigui la CPU el qui faixi tota la feina. El model utilitzat és un Broadcom® iSCSI 2 WY733. Podem trobar el manual a la pàgina web de Dell [13] i [18] ja que és un dispositiu fabricat exclusivament per aquesta marca.

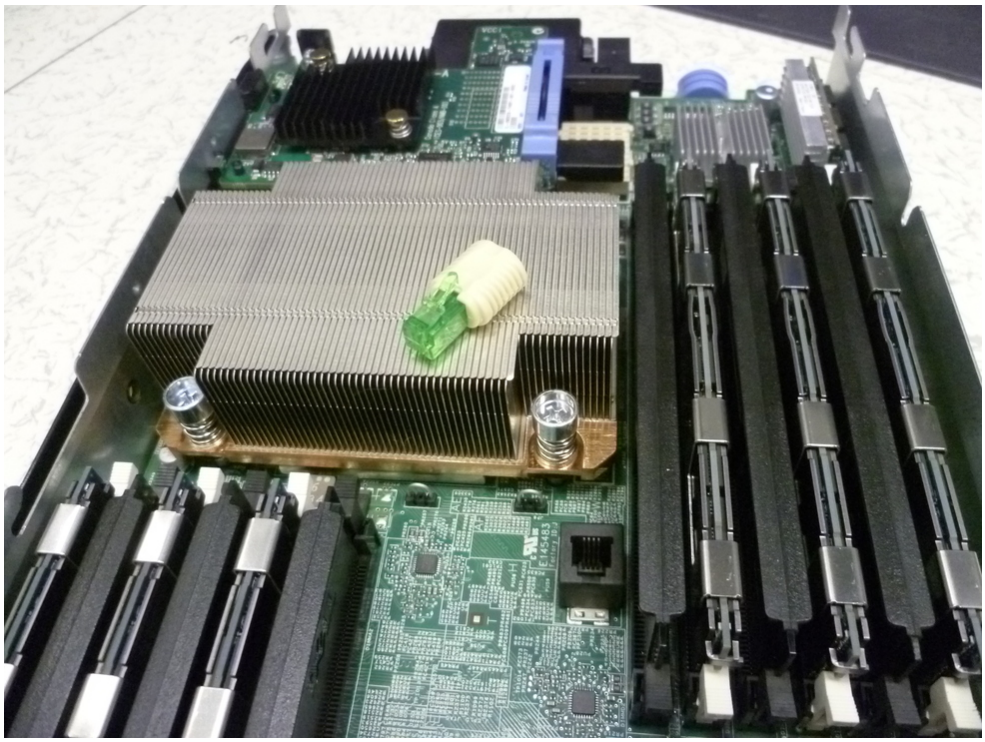


Figura 31: Dispositiu iSCSI Offload Engine

Memòria RAM

Els tipus de memòria que utilitza cada conjunt de nodes és diferent. S'utilitza una arquitectura de memòria compartida distribuïda detallada en el capítol 8.1 Estudi de l'arquitectura de hardware.

Els nodes M605 utilitzen RAM convencional DDR2 a 800Mhz.

Els nodes M610 utilitzen RAM convencional DDR3 a 1333Mhz,

Els nodes M600 (Intel® E5410 i node màster) fan servir FB-DIMM a 667Mhz (Figura 32).

La FB-DIMM no és compatible amb la RAM convencional.

Aquest nom significa “Fully Buffered DIMM” i utilitza una tecnologia que pretén millorar la fiabilitat i la densitat del sistema de memòria. En memòries convencionals el controlador de memòria ha d'estar connectat físicament a cada línia de dades de cada mòdul DRAM mitjançant per exemple un bus multidrop, on tots els components de la placa són connectats al mateix bus i un procés de sincronització i arbitre dona permís per enviar dades. Això sol limitar la freqüència de treball al voltant d'uns 200-400Mhz i per tant en limita el rendiment. Les memòries FB-DIMM pretenen evitar aquest problema incorporant un bus anomenat AMB per Advanced Memory Buffer entre el controlador de memòria i el mòdul de memòria. A diferència dels busos paral·lels de les DRAM convencionals, la FB-DIMM disposa d'una connexió sèrie entre aquest bus i el controlador que permet també incrementar l'amplada física dels mòduls DRAM sense haver de modificar el

controlador de memòria. D'altra banda la FB-DIMM no escriu directament al controlador sinó que ho fa al buffer AMB que pot realitzar tasques com la compensació de deteriorament del senyal, emmagatzemant les dades i tornant a enviar el senyal. Pot oferir també correcció d'errors evitant així la sobrecàrrega del processador o el controlador de memòria central, i per tant en una arquitectura multi-processador i multi-nucli reduint notablement aquesta càrrega. També pot detectar camins físics deteriorats i evitar-ne el seu ús o gestionar l'ordre en que s'envien els senyals al controlador de memòria per facilitar-li la tasca de funcionar en paral·lel. Finalment un altre avantatge és que aquesta tecnologia pot utilitzar controladors de memòria que no coneguin quin tipus de memòria s'ha inserit (DDR2, DDR3, etc.) sinó que només es comuniquin amb l'AMB.

Les desavantatges d'aquesta tecnologia són que cada petició d'accés a la memòria sofreix una latència, requereix més energia i la velocitat d'escriptura disminueix. En entorns de computació d'alt rendiment com el nostre en que es requereix un accés elevat a memòria, pot suposar una pèrdua notable de rendiment. Un altre desavantatge és que els mòduls de FB-DIMM són més cars que els de memòria convencional.

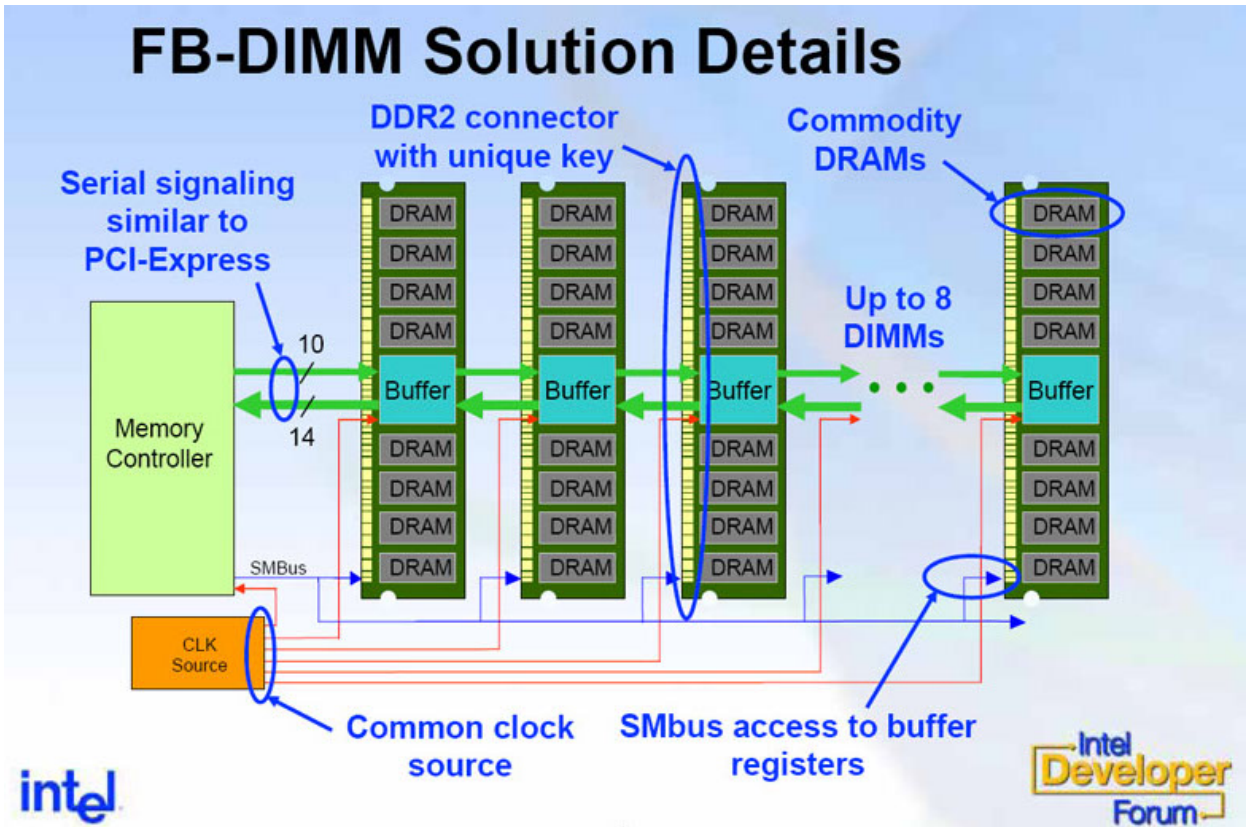


Figura 32: Esquema de la tecnologia FB-DIMM. Transparències de Intel®.

Per veure més al respecte, consultar les fonts [19],[20] i [21].

Processadors

Els processadors dels que disposen els nodes són l'Intel® Xeon® E5410, AMD Opteron™2356 i Intel® Xeon® E5645.

Cada node disposa de dos sockets i de dos processadors del mateix tipus instal·lats. Els tipus de processador es detallen a la Taula 7:

	Intel® Xeon® E5410	***AMD Opteron™ 2356***	***Intel® Xeon® E5645***
Freqüència	2.33 Ghz	2.30 Ghz	2.40 Ghz
Nuclis	4	4	6
Fils d'execució	4	8	12
Caché L1	4x128KB	4x64KB	6x64KB
Caché L2	2x6144 KB compartits	4x512 KB	6x256KB
Caché L3	N/A	2048KB compartits	12288KB compartits
Bits d'adreces	38 físiques, 48 virtuals	48 físiques, 48 virtuals	40 físiques, 48 virtuals
Mida del TLB	-	1024 pàgines de 4K	-
FSB(Mhz), QPI o HT (GT/s)	1333 MHz (FSB)	1.00 GT/s (HyperTransport)	5.86 GT/s (QPI)
Controlador de memòria	Hub separat del chip, compartit en sockets (SMP). 4 canals compartits en 2 sockets	Integrat, 2Ghz (NUMA)	Integrat, 3 canals DDR3 per socket (NUMA)
Socket	LGA771	Socket F	LGA1366
Tecnologia	45nm	65nm	32nm
Consum PIC	80W	75W	80W
Velocitat memòria	667Mhz	800Mhz	1333MHz

Taula 7: Característiques dels processadors dels nodes

4.1.3 Servidors XFire i Vega

Els altres dos servidors de càlcul dels que disposa CIMNE són els anomenats XFire i Vega. Ambdós són de la ja inexistent marca SUN Microsystems (fou comprada per Oracle el 2009), en format torre i estan situats al rack 2, que és compartit entre diversos servidors i alimentat per les línies 1 i 2 de 16 amperes cadascuna.

Disposen de les següents característiques:

Vega

Model:	SUN Ultra 40 M2 Workstation
CPU:	2 x Dual-Core AMD Opteron™ 2214 [25]
Memòria:	32Gb DDR2-667 DIMM, en 8 mòduls de 4Gb (4 mòduls * socket)
Disc:	1 x Hitachi Deskstar™ 250Gb SATA 3.0Gbps 7200RPM
Font d'alimentació:	1x1000W
Xarxa:	2 x Ethernet 1Gb
Manual i especif.:	veure ref. [22],[doc12],[doc13]

Les diferències del model Sun Ultra 40 M2 amb el Sun Ultra 40 són que utilitza el socket F enlloc del socket 940, memòries DDR2-667 amb mòduls de fins a 4GiB ECC enlloc de PC3200 i mòduls de fins a 2GiB ECC, capacitat per 8 discs SATA o SAS enlloc de 4, i no disposa de PCIe x4 sinó de PCIe x8.

Podem veure una imatge de la part frontal de Vega a la Figura 33 i una de l'interior a la Figura 34 on s'aprecien les dues CPU amb els 4 mòduls de memòria RAM associats cadascuna.

Figura 33: Sun Ultra 40 M2, Vega vist des del frontal

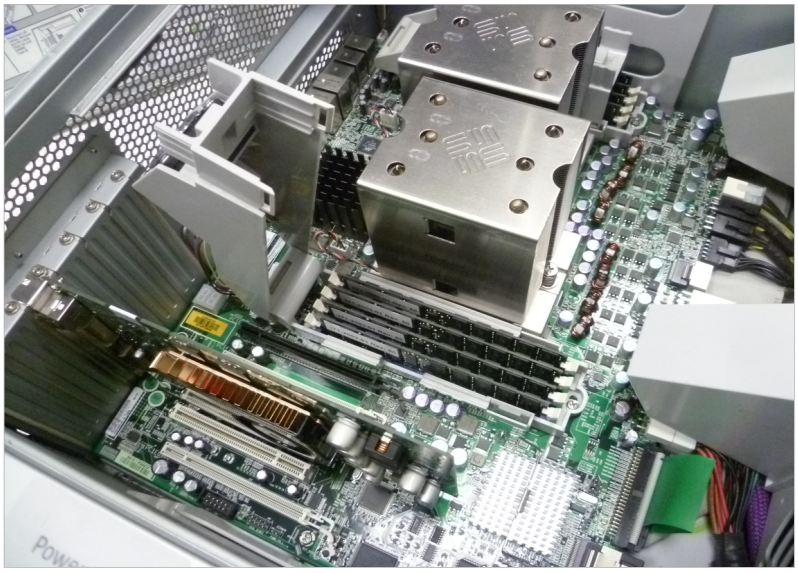


Figura 34: Interior del Sun Ultra 40M2 (Vega)

XFire

Model:	SUN Fire X4600
CPU:	4 x Dual-Core AMD Opteron™ 885 [24]
Memòria:	32 GB DDR1 400 (PC3200) DIMM, en 16 mòduls de 2Gb.
Disc:	2 x Fujitsu 74GB SAS 10.000RPM
Xarxa:	4 x Ethernet 1Gb
Font d'alimentació:	4 x 950W
Manual i especif.:	Veure ref. [23],[doc10],[doc11]

La diferència entre un X4600 i un del model X4600 M2 és que primerament el model X4600 només pot allotjar processadors de la sèrie 800 de AMD (socket 940) que són Dual Core i que únicament suporten memòria DDR1. El model M2 pot allotjar processadors de socket F, per exemple els Barcelona que són quad-core i suporten RAM PC2-5300 DDR2.

A més els mòduls de CPU del M2 poden allotjar 4 o 8 DIMMS mentre que en l'anterior model només 4. Es pot diferenciar un mòdul de CPU de l'altre mitjançant els colors de les ranures DIMM: si són alternades en blanc i negre es tracta del model més senzill, mentre que dos blancs i dos negres consecutius s'utilitzen en el model M2.

A la Figura 35 podem veure dos dels quatre ventiladors extrets a XFire. Segons les especificacions cadascun pot arribar a un consum pic de 108W i 4.5A.

A la Figura 36 tenim una vista general de l'interior i especialment de la configuració amb 4 mòduls de CPU. A la Figura 37 veiem un d'aquests 4 mòduls amb els seus 4 DIMM de RAM DDR1.

Figura 35: Ventiladors Fire X4600

Figura 36: Mòduls de CPU i placa base Fire X4600

Finalment a la Figura 38 hem extret una font d'alimentació de les 4 de les que disposa XFire.

Figura 37: Mòdul de CPU amb 4 DIMM RAM Fire X4600

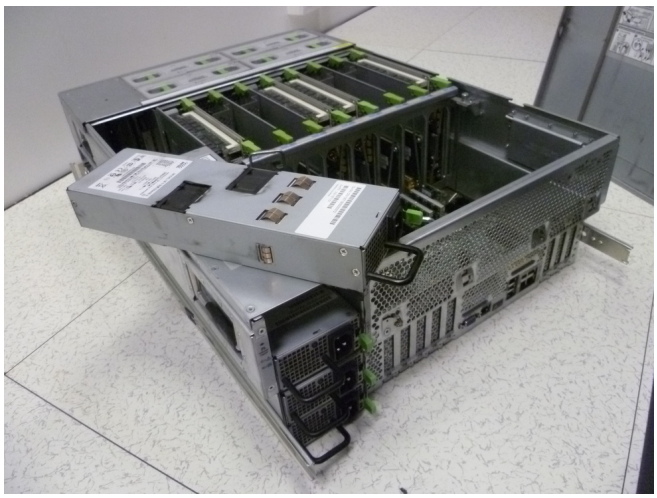


Figura 38: 4 fonts d'alimentació Fire X4600

Taula comparativa dels processadors entre XFire i Vega

Mostrem a continuació la Taula 8 que compara les característiques dels processadors dels servidors XFire i Vega.

	AMD Opteron™ 2214	***AMD Opteron™ 885***
Freqüència	2.20 Ghz	2.60Ghz
Nuclis	2	2
Fils d'execució	2	2
Caché L1	2 x 64 KB 2-way associative instruction 2 x 64 KB 2-way associative data	2 x 64 KB 2-way associative instruction 2 x 64 KB 2-way associative data
Caché L2	2x1048KB exclusiu 16-way associative	2x1048KB exclusiu 16-way associative
Caché L3	N/A	N/A
Bits d'adreces	Fins a 1TiB físic / 256TiB virtual	Fins a 1TiB físic / 256TiB virtual
HyperTransport	1000 MHz (1GT/s)	1000 Mhz (1GT/s)
Controlador de memòria	Integrat al chip, DDR2 PC2-5300 (NUMA)	Integrat al chip, DDR PC3200 (NUMA)
Socket	Socket F/1207	Socket 940
Tecnología	90nm	90nm
Consum PIC	95W	95W

Taula 8: Comparació dels processadors de XFire i Vega

4.1.4 Comunicacions

Disposem de dos tipus de connexions per les comunicacions del servei de càlcul, una és Ethernet 1Gb i l'altre Infiniband.

Infiniband

La connectivitat Infiniband és realitza en el clúster mitjançant el switch Cisco M SFS7000E DDR 4x connectat com a mòdul C1 del chassis. Cada node disposa d'una targeta de xarxa (anomenada també HCA, Host Channel Adapter) Infiniband DDR de dos ports i de 2.5GT/s o 5GT/s en el cas dels M610.

En casos com l'ethernet habitualment es fan servir sockets POSIX per la comunicació proporcionant un espai de memòria virtual accessible a nivell d'usuari. Quan un usuari vol transmetre o rebre dades de la xarxa ha de llegir o escriure en aquest socket, fet que implica que el kernel ha d'involucrar-se en les comunicacions copiant les dades d'espai de kernel a espai d'usuari o viceversa i per tant afegint una gran latència a la comunicació. Això no és positiu especialment per el món HPC on tot ha de ser el més ràpid possible. [26]

Per altra banda i més properament al hardware, les dades enviades o les rebudes s'han de passar d'espai de kernel al buffer de la targeta de xarxa, creant així un pas més que encara introdueix més latència. En transferències molt grans o constants aquest factor pot ser d'influència en el rendiment general.

La tecnologia Infiniband evita aquest problema fent servir el que s'anomena RDMA: Remote Direct Memory Access [27], metodologia que permet que un node pugui accedir a la memòria d'un altre node sense haver de passar per el sistema operatiu. La targeta de xarxa té permís per accedir a zones de memòria RAM d'aplicacions d'usuari i per tant evita el pas de copiar al buffer, a l'espai de memòria de kernel i després al d'usuari. Evita el problema de la redundància de còpies, del temps de canvis de context de la CPU, caches, etc.

Com a topologia s'utilitza un paradigma diferent al d'Ethernet. En aquest cas es sol formar una topologia “switched fabric topology” en que cada node és connectat a tot altre node amb una o més connexions punt a punt. Els tipus de switch que implementen aquesta topologia s'anomenen “crossbar switches” (Figura 39). És tot el contrari de les tropologies de broadcast que es feien servir en els hubs ethernet.

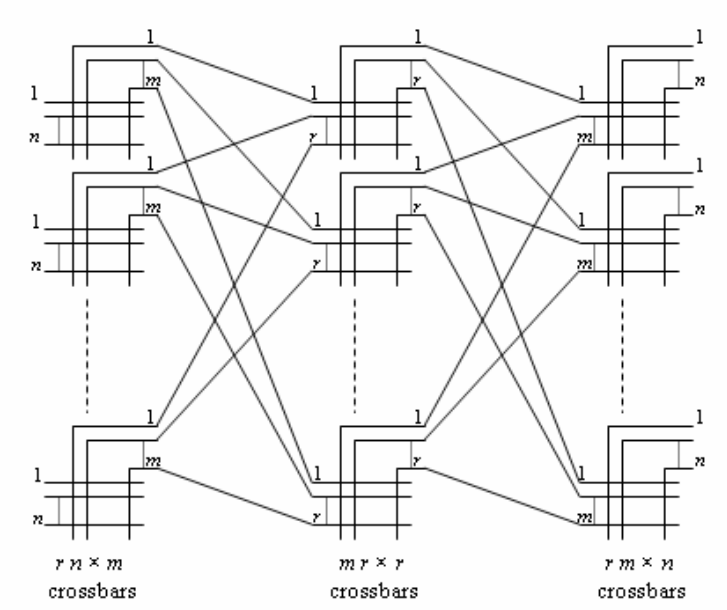


Figura 39: Switchs crossbar en topologia Clos

El SFS 7000E de CISCO utilitza una topologia Fat Tree (Figura 40): Tots els HCA es connecten a la xarxa a les fulles d'un arbre. Cada fulla és un switch Infiniband i està connectat a un switch arrel. En el nostre cas realment funciona com un switch normal ja que només disposem d'un switch.

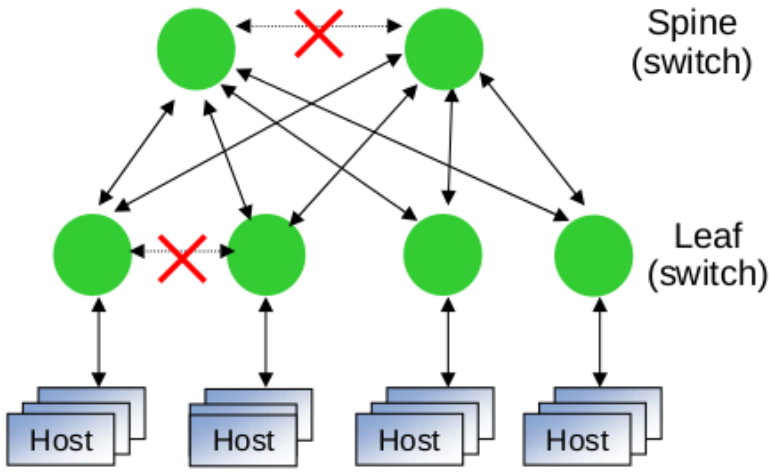


Figura 40: Topologia Fat Tree

El switch del que disposem és de doble rati de senyalització, DDR, i quatre fils per cable, 4x.

El rati de senyalització indica la velocitat de transmissió en Gbps a la que s'efectua la transmissió. Per un DDR la velocitat està en 5Gbps per cada fil. DDR a més fa servir una codificació 8Bits/10Bits – cada 10 bits enviats, 8 bits són de dades – , i per tant té una eficiència del 80%.

Si cada cable porta 4 fils (4x), i cada fil assoleix 5Gbps obtindríem un màxim teòric de 20Gbps a cada enllaç, no obstant el 80% de rendiment degut al a codificació redueix aquests 20Gbps a 16Gbps efectius. [26]

A la Taula 9 mostrem una comparativa de les diferents velocitats de transmissió efectives assolides per altres tipus d'Infiniband [doc15].

	SDR	***DDR***	***QDR***	***FDR-10***	***FDR***	***EDR***
1x	2 Gbit/s	4 Gbit/s	8 Gbit/s	10,3 Gbit/s	13,64 Gbit/s	25 Gbit/s
4x	8 Gbit/s	**16 Gbit/s**	32 Gbit/s	41,2 Gbit/s	54,54 Gbit/s	100 Gbit/s
12x	24 Gbit/s	48 Gbit/s	96 Gbit/s	123,6 Gbit/s	163,64 Gbit/s	300 Gbit/s

Taula 9: Comparació de velocitats de transmissió Infiniband

Hem de tenir en compte però, que la limitació imposada en quant a la velocitat de transmissió podria arribar a ser del bus PCIe. Com hem vist a l'apartat 4.1.2.2, Targetes de xarxa les especificacions de les HCA dels M600 i M605 es connecten a un bus PCIe 2.0 de 2.5GT/s, mentre que els M610 poden connectar-se a un PCIe 2.0 de 5GT/s. Els busos PCIe bàsics de un sol enllaç disposen de dos “fils” ja que la comunicació és bidireccional, suportant cadascun 2.5Gbps i per tant una transferència de dades màxima de 5Gbps. Si el bus PCIe de la placa base fos 1x, ens trobaríem en que el coll d'ampolla seria aquest bus i no la HCA. No obstant, si mirem les especificacions dels nodes M600, M605 i M610 veurem que disposen d'un bus PCIe x8. El bus x8 multiplica per 8 el nombre d'enllaços aconseguint un total de 40Gbps, i si tenim en compte la codificació 8B/10B que s'utilitza en els PCIe, la velocitat de transmissió efectiva serà de 64Gbps. Podem constatar per tant que el bus no és el coll d'ampolla d'aquesta comunicació [30], [doc9]

Per acabar aquest apartat hem de dir que el protocol Infiniband no treballa amb la mateixa pila de protocols que l'arquitectura Ethernet.

La Infiniband és una xarxa gestionada de forma central i per això necessita un àrbitre en algun punt. Figura 41.

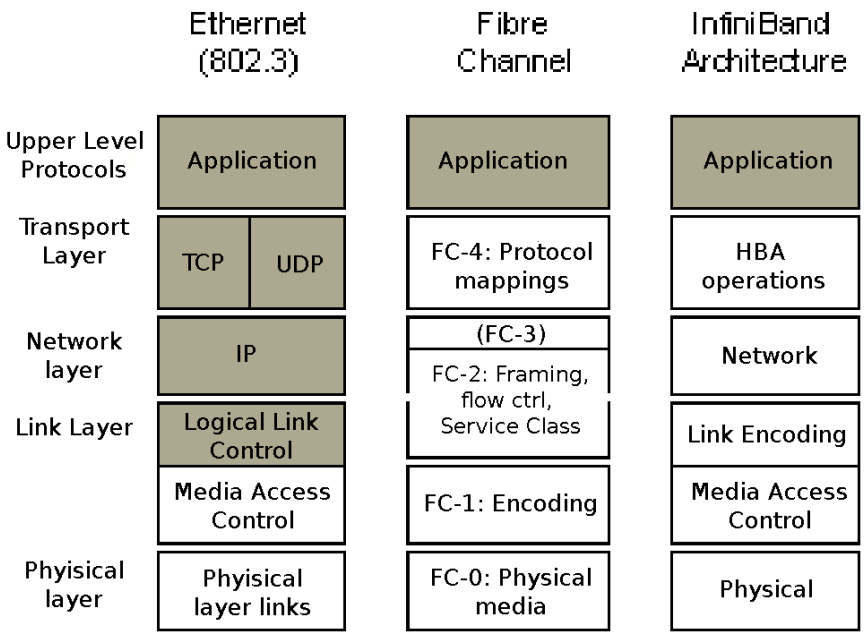


Figura 41: Comparació arquitectures de xarxa

Aquest àrbitre serà un dimoni anomenat Subnet Manager (SM) que s'executarà en algun node i que realitzarà les següents funcions:

- Escaneig de la xarxa: descobrir nous HCA, encaminament, gestió dels ports
- Monitoreig de la xarxa: inserció/eliminació de nous dispositius, canvis d'estat, comptadors de ports, etc.
- Manteniment i informació de la xarxa: Particions, grups multicast, serveis, camins, etc.

L'existència d'un dimoni SM és indispensable per el funcionament de la xarxa i pot funcionar en un switch o en un node.

Per més informació respecte la tecnologia Infiniband consultar el document intern [doc8].

Ethernet

La Infiniband fins al moment és exclusiva pels nodes del cluster ja que no disposem de HCAs a XFire ni a Vega. En aquests dos servidors es disposen de targetes de xarxa integrades de 1Gb, de igual forma que passa amb tots els nodes en el Fabric A, per tant tots els servidors dels que disposem tenen targetes controladores de com a mínim 1Gb.

Com a suport a aquestes dispositius disposem d'un switch Dell™ PowerConnect™ 5424 situat al rack 2, el mateix del clúster, Figura 42. Aquest switch és utilitzat exclusivament per la interconnexió de nodes del clúster i que disposa de capacitats de gestió eficient del protocol iSCSI. A més és pot gestionar per interfície web, tot i que fins al moment no ha estat configurat.

Per altra banda al rack del clúster disposem de 4 punts de xarxa que arriben del rack de comunicacions del CPD: dos punts estan a la xarxa interna VLAN192, i altres dos punts a l'àrea desmilitaritzada de CIMNE (VLAN147) i per tant possibiliten el disposar d'una IP pública pel node màster del clúster.

Només hi ha connectivitat entre XFire, Vega i Acuario mitjançant la xarxa VLAN192, i en cap cas es pot arribar als nodes de forma individual sense passar per Acuario.

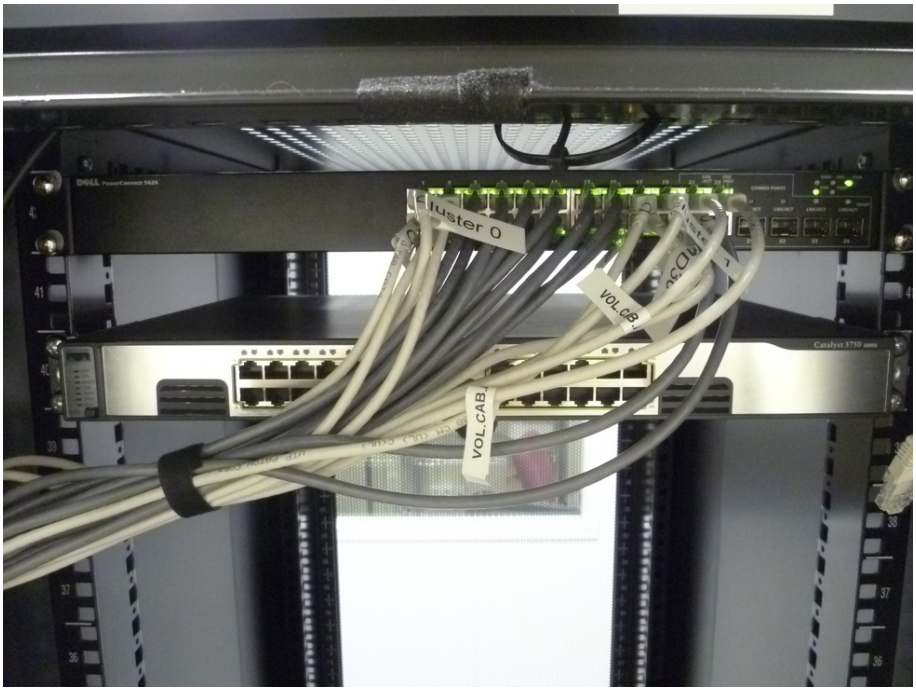


Figura 42: Switch Dell™ PowerConnect™ 5424 (superior)

4.1.5 Emmagatzemament

Com a opció d'emmagatzemament s'optà per adquirir una cabina de discs de tipus SAN (Storage Attached Network) amb dues interfícies ethernet iSCSI i capacitat per allotjar fins a 16 discs durs SAS o SATA2. Es tracta del model Dell™ PowerVault™ MD3000i que passem a descriure a continuació.

4.1.5.1 Descripció física

La cabina disposa a la part frontal de 16 ranures per col·locar-hi discs durs SAS o SATA2. Aquests discs durs han d'anar acompanyats d'un porta-discs estàndard de Dell, del mateix tipus que s'empra en els servidors (Figura 44). A més per discs SAS els porta-discs disposen d'un convertidor de connector per fer que encaixin de la mateixa forma que els SATA2. Els discs que es poden inserir han de ser certificats per Dell (discs de servidor), per tant no és possible comprar un disc dur de qualsevol fabricant i afegir-lo a la cabina ja que rebrem l'error que apareix a la Figura 43.

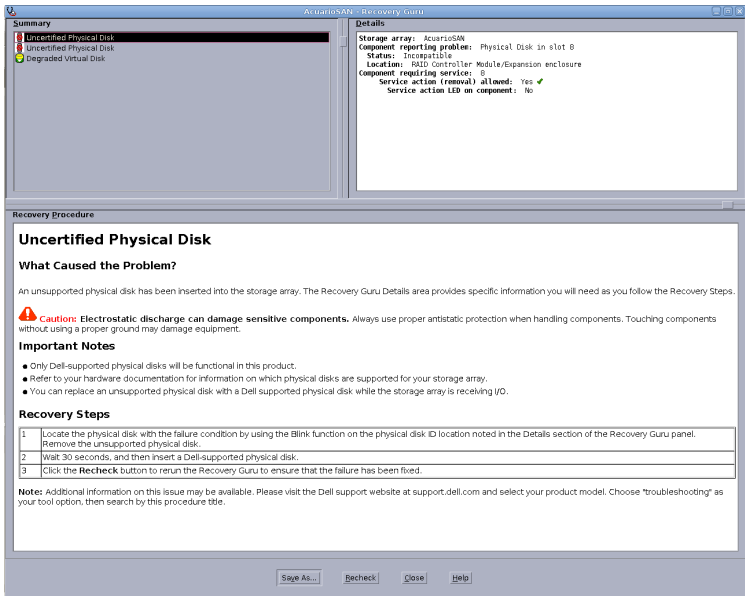


Figura 43: Missatge d'error al col locar discs no certificats per Dell al SAN

Figura 44: Part frontal de la cabina de discs, amb un disc tret a fora

A la part posterior de la cabina hi trobem dues fonts d'alimentació i un mòdul de connexió. Aquest mòdul disposa de 5 ports (Figura 45):

- 2 ports ethernet iSCSI, per transmissió de dades i exportació de volums iSCSI. No són ports amb balanceig de càrrega sinó que s'utilitzen per redundància. Es poden assignar diferents ports a diferents volums.
- 1 port ethernet de gestió, per accés a la subxarxa de gestió de la cabina.
- 1 port STK, ofereix la possibilitat de connectar en cascada diverses cabines de discs.
- 1 port sèrie PS2 de gestió per utilitzar en casos de recuperació.

En models posteriors es poden instal·lar mòduls de connectivitat per fibra òptica o SAS, però no és el cas d'aquesta cabina.

Figura 45: Mòdul posterior de la MD3000i

4.1.5.2 Gestió de la Dell™ PowerVault™ MD3000i

La gestió de la cabina es realitza mitjançant el programa propietari Dell MD3000i StorageManager, aplicació Java que s'ha d'instal·lar a l'equip que tingui connectat el port de gestió (Figura 46).

El funcionament de la gestió és com qualsevol altre cabina. Es poden definir volums, diversos sistemes RAID, donar accessos, autenticacions, etc.

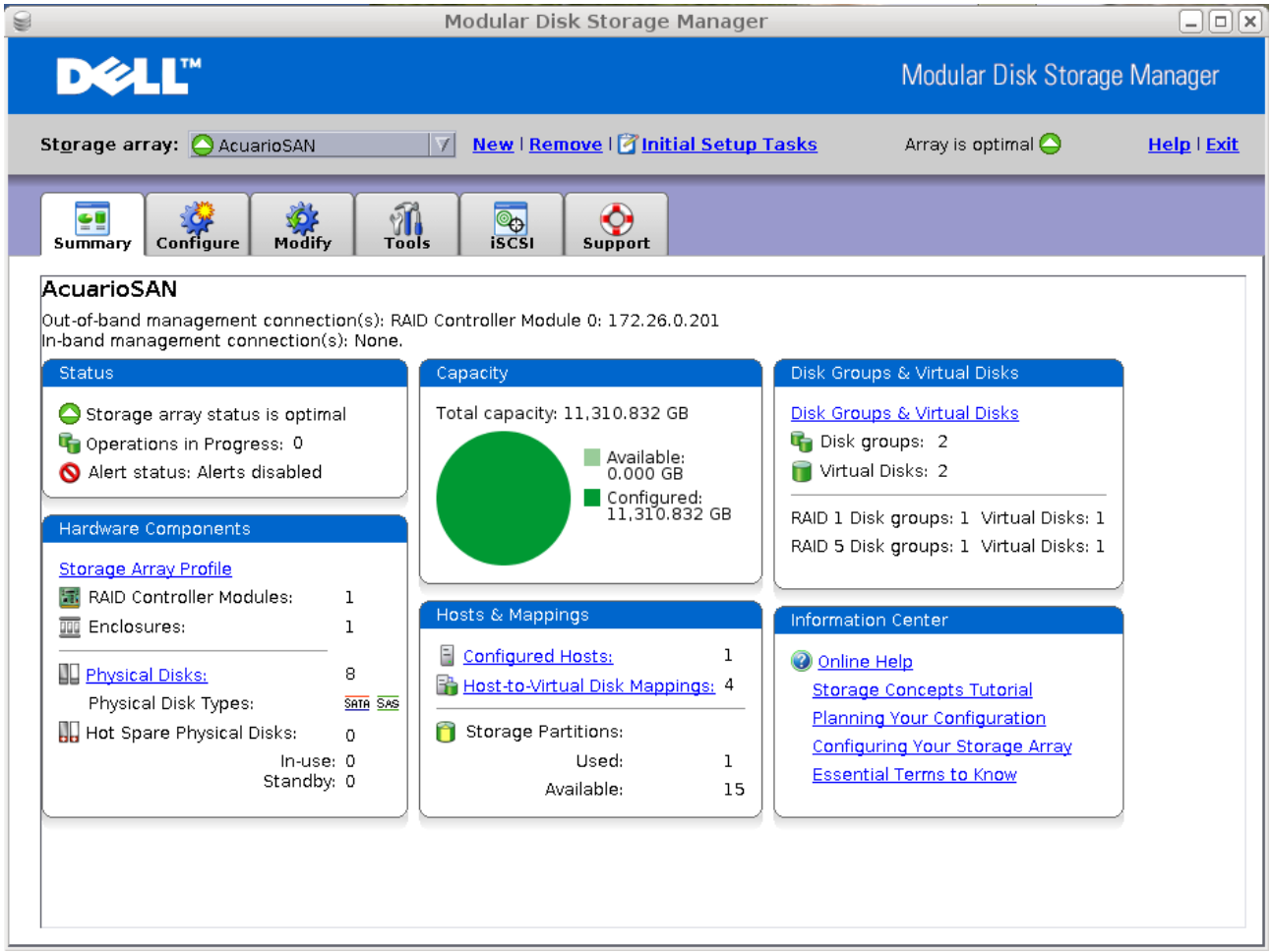


Figura 46: Gestió de la MD3000i

4.1.6 Visió general de la infraestructura

Mostrem a la Figura 48 i a la Figura 47 dues imatges il·lustratives dels elements que componen el clúster de càlcul de CIMNE.

Figura 48: Part frontal del clúster i la cabina

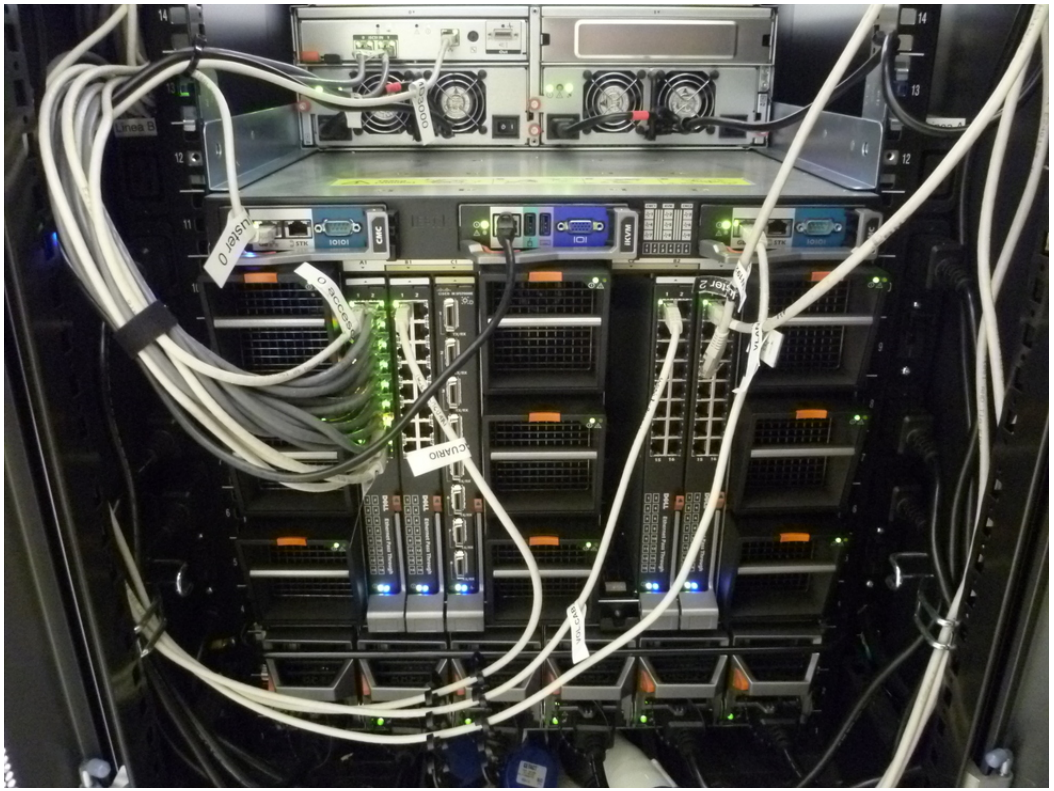


Figura 47: Part posterior del clúster i la cabina

4.2 Estat de l'art

A continuació analitzem l'estat actual de la tecnologia i les alternatives que se'ns ofereixen en aquest àmbit per tal d'escollir la solució més adequada a les nostres necessitats.

4.2.1 Definició de clúster

Quan parlem de clusters en general hem de determinar de quin tipus ho feim. Hi ha actualment dues definicions per “clúster” que hem de conèixer per no confondre'ns a l'hora de cercar informació, ja que en molts casos es pot no estar parlant del que esperem.

- Clúster d'alta disponibilitat (*High Availability*): Aquesta definició fa referència a conjunts d'ordinadors servidor connectats entre sí que pretenen proporcionar un servei robust, estable, i sempre disponible utilitzant principalment la redundància. No és el cas que ens ocupa.
- Clúster de càlcul (*High Performance*): Aquesta definició és la que interessa per aquest treball i fa referència a conjunts d'ordinadors que treballen paral·lelament o com un de sol per tal de proporcionar potència de càlcul i un major rendiment per els càlculs a processar.

Dins dels clústers de càlcul trobem diferents classificacions que a la literatura a vegades no coincideixen, per tant el que descrivim a continuació pot no ser exacte:

- Beowulf, conjunt de màquines habitualment idèntiques, barates, estàndards i fàcils d'adquirir que són connectades mitjançant una xarxa, disposen d'un node principal que les controla i que proporcionen la capacitat de realitzar càlculs en paral·lel. Normalment els nodes secundaris no tenen ni teclat ni pantalla i es caracteritzen per comptar amb hardware estàndard i no propietari a més de poder-se implementar amb sistemes operatius Unix estàndard. Inicialment la idea sorgí de Thomas Sterling i Donald Becker el 1994 [33] mentre treballaven per la NASA i pretengué utilitzar sistemes d'escriptori per tal de minimitzar costos. Avui en dia l'arquitectura de clúster Beowulf és utilitzada en molts centres de computació tot i que no fent servir hardware barat o no propietari.
- Clúster de memòria distribuïda, habitualment es tracta d'un cluster Beowulf però amb hardware propietari on es prioritza el rendiment al preu. Es tracta del clúster de CIMNE.
- Supercomputador, són els precursors dels antics mainframes sorgits el 1960 de la mà de Seymour Cray [34] amb el CDC6600 i són infraestructures que compten habitualment amb milers de processadors i grans quantitats de memòria, habitualment propietaris (p.ex. IBM, HP, Cray, etc.) connectats amb xarxes d'alt rendiment i que proporcionen una gran capacitat de càlcul. Requereix sistemes operatius específics i optimitzats a diferència dels Beowulf [35].
- Grid computing, hi ha diverses definicions, però totes coincideixen amb que és una xarxa de computadors heterogènia distribuïda amb alguna xarxa de llarg abast com pot ser Internet i que pot utilitzar-se per diverses tasques, ja sigui computació, investigació, xarxes de sensors, etc. Els punts de la definició de Ian Foster indiquen tres característiques principals [36]:
  - Els recursos de computació no estan administrats en un lloc centralitzat
  - S'utilitzen estàndards oberts
  - S'aconsegueix qualitat de servei

4.2.2 Arquitectures de hardware

En el món de la computació d'alt rendiment trobem sovint diverses arquitectures de hardware que venen motivades per la necessitat d'especialitzar els computadors per tasques de determinat tipus.

Per exemple en un model de simulació i predicció del temps es necessita emmagatzemar una gran quantitat de variables en memòria per cada fragment geogràfic. Per cada km² de terra i per cada instant de temps hi ha paràmetres de temperatura, pressió, humitat, vent, etc. Cadascun d'aquest km² requereix interactuar amb tot altre km² adjacent a ell i per tant es necessita tenir en memòria gairebé tota la malla de terra per cada instant. Com més resolució es tingui, per exemple en lloc de km² utilitzant m², més encertat serà el model resultant. Tota aquesta informació emmagatzemada en memòria necessita ser accedida, llegida i escrita mitjançant càlculs simples, i per tant necessitarem un ordinador amb una memòria principal molt gran i ràpida. Seria un exemple d'utilització d'un clúster de memòria compartida on és més important la memòria que el processador.

En altres casos ens interessarà donar protagonisme al processador de forma individual, i en altres es podrà dividir el codi en fragments que s'enviïn a calcular a diferents parts del clúster per finalment unir-ne els resultats.

Veiem a continuació els tipus d'arquitectures principals.

4.2.2.1 Taxonomia de Flynn

La taxonomia proposada per M.J. Flynn el 1966 ha estat fins el moment el punt de partida en el camp de la computació. Es basa en la noció d'instrucció i dades que poden ser simultàniament manipulades per una màquina. Flynn classifica els tipus de computadors en quatre categories.

Per la següent explicació entendrem com a “stream” una seqüència de instruccions o de dades.

La major part de la informació d'aquest apartat ha estat obtinguda del document intern [doc14].

SISD – Single Instruction Stream, Single Data Stream

- Processador tradicional d'arquitectura von Neumann, Figura 49.
- Una unitat de control (CU) pren una sola instrucció de memòria (IS)
- La IS fa que la CU generi les senyals adequades per tal que l'element de procés (PE) operi amb una sola dada (DS) que obtindrà de memòria (una PE pot ser p.ex. la ALU).
- Es correspon amb l'arquitectura de PC que hi havia fins a meitat de la primera dècada del 2000, on els processadors eren de un sol nucli, o amb els antics mainframes.

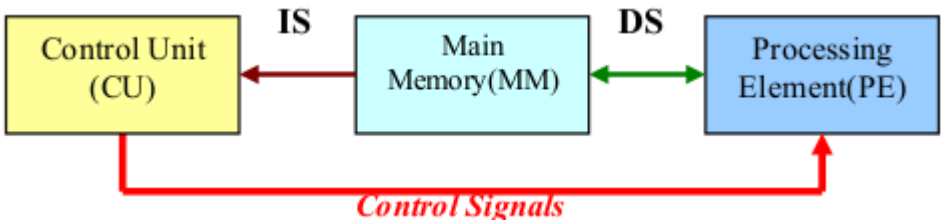


Figura 49: Esquema arquitectura SISD

SIMD – Single Instruction Multiple Data

- L'arquitectura SIMD sorgeix als inicis del paral·lelisme. Anomenats també processadors vectorials.
- Model utilitzat en la tecnologia MMX dels Pentium, instruccions SSE, GPUs, i en general usat en computadors on els càlculs són molt estructurats, per exemple en el processat d'imatge, matrius i 3D.
- La unitat de control CU pren una instrucció IS de memòria, la descodifica i envia les senyals corresponents a totes les unitats de procés PE_i , Figura 50.
- Cada PE_i executa la mateixa instrucció però obtenint dades de diferents posicions de memòria.
- Per exemple usat en les ocasions en que es disposa d'una matriu i un vector grans on s'ha d'aplicar una mateixa operació a cada element.

  Posem Y = A*X, on X,Y són vectors de n elements i A és una matriu de NxN. L'operació es formalitza com:

$$y_i = \sum_{j=1}^{N} a_{ij} * x_j \text{ for all i} = 1, 2, 3, \ldots, N$$

  Llavors si tenim de PE_1 a PE_n podrem donar a cada PE_i una copia de la fila *i* de la matriu A i el vector X i fer que executin de forma síncrona la operació de suma. Això produirà un vector resultant Y en N passos enlloc de fer-ho amb N^2 que ocorreria si es fes de forma seqüencial.

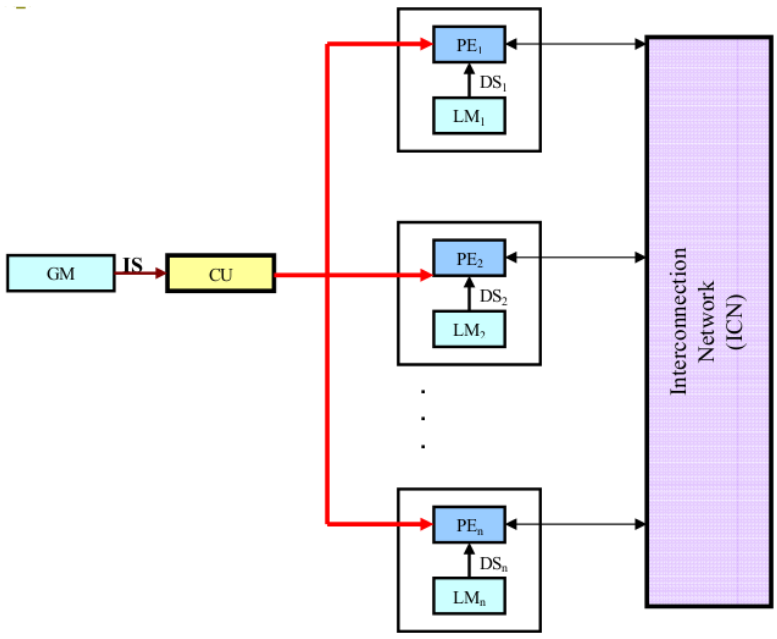


Figura 50: Arquitectura SIMD

MISD – Multiple Instruction Single Data

- No s'empra a la pràctica excepte en casos molt concrets on es cerca una alta tolerància a falls. Per exemple l'ordinador de control de les naus espacials.
- Múltiples instruccions actuen sobre un mateix conjunt de dades.
- Podria ser útil en el desxifrat de missatges codificats usant diferents algorismes, en l'ús de múltiples filtres de freqüència en una mateixa senyal, etc.

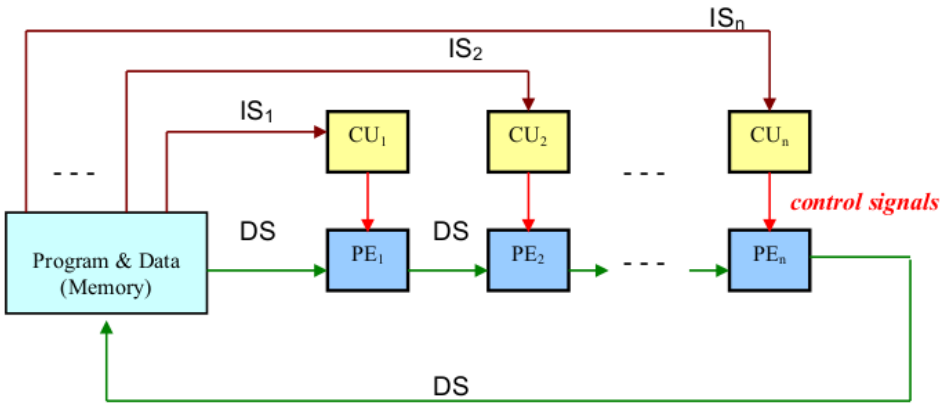


Figura 51: Arquitectura MISD

MIMD – Multiple Instruction Multiple Data

Aquesta és l'arquitectura més interessant per el projecte que ens ocupa i la que realment s'està fent servir en la majoria dels computadors actuals. Disposa de diverses variants que veurem un cop entès el concepte general.

- Diversos processadors PE tenen assignat cadascun un conjunt de dades i executen el seu propi conjunt d'instruccions a sobre d'aquest.
- Es tracta d'un multi-processador real.
- Són màquines de propòsit general, a diferència dels SIMD.
- Explota el paral·lelisme asíncron.
- Es proporciona una xarxa d'interconnexió (ICN) per la comunicació entre processador-processador i processador-memòria.
- Quan tots els processadors executen el mateix programa (que no mateix codi), anomenem el càlcul com Single Program Multiple Data (SPMD).
- Es fa servir en servidors, en PC multiprocessadors (a partir de l'any 2005), clústers, etc.

A la Figura 52 veiem en general com s'estructura un computador d'aquest tipus. No obstant dins la classificació MIMD existeixen tres classificacions més que definiran realment la gran part dels computadors utilitzats actualment tant per càlcul com per propòsit general.

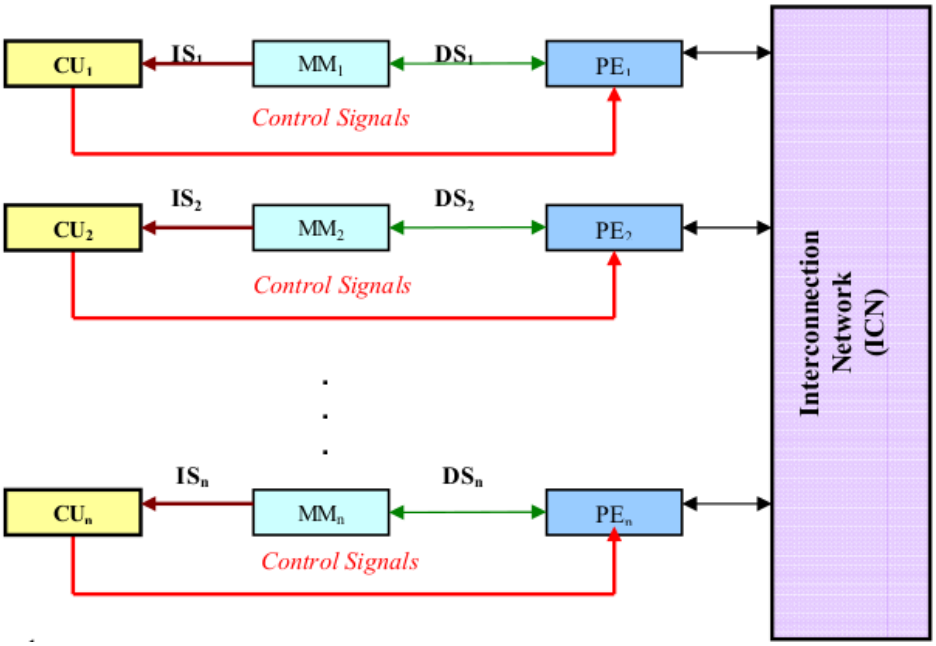


Figura 52: Arquitectura general de MIMD

La classificació d'arquitectures MIMD es basa en com els processadors es comuniquen amb els altres. Tenim tres categories.

1. **Memòria distribuïda**

- Existeix un espai de memòria separat per cada processador: una mateixa adreça generada per dos o més processadors es refereix a diferents elements de memòria.
- Un processador no té permís per accedir al mòdul de memòria d'un altre processador, sinó que s'ha de comunicar amb l'altre mitjançant el pas de missatges.
- S'anomenen arquitectures NO Remote Memory Access (NORMA).

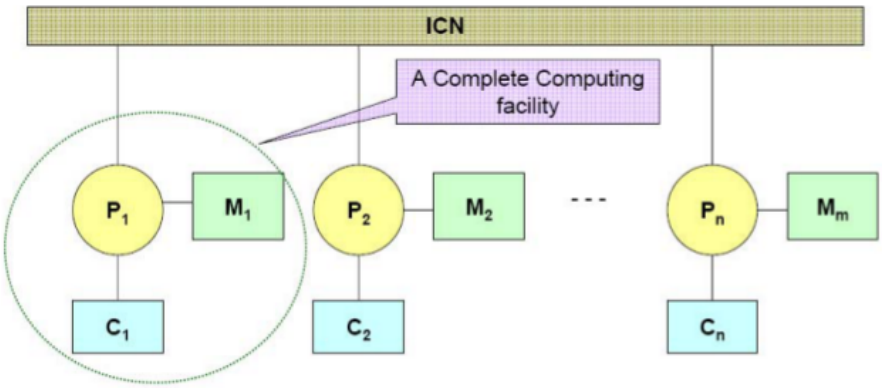


Figura 53: Arquitectura MIMD de Memòria Distribuida (NORMA)

2. **Memòria compartida**

- Hi ha un sol espai de memòria compartit amb tots els processadors: una adreça generada per diferents processadors apunta al mateix element de memòria en un dels mòduls de memòria.
- Existeix un bus entre la memòria i els processadors i per tant poden haver-hi problemes de latència si tots hi accedeixen al mateix temps.
- El primer pas del processador es cercar en la pròpia cache, si no es troba la dada llavors s'accedeix a memòria.
- Quan el processador guanya accés al bus pot accedir a qualsevol mòdul de memòria de forma uniforme. D'aquí el nom de Uniform Memory Access: UMA.
- Un altre nom que els defineix és Symmetric Multi Processor (SMP). Un SMP es caracteritza per:
  - Processadors de idèntiques funcionalitats
  - Idèntic accés als recursos per tots els processadors
  - El Kernel del S.O. Pot córrer en qualsevol processador
- En arquitectures actuals, on hi ha una placa base amb un processador i diversos nuclis, els nuclis es comporten internament com a SMPs.

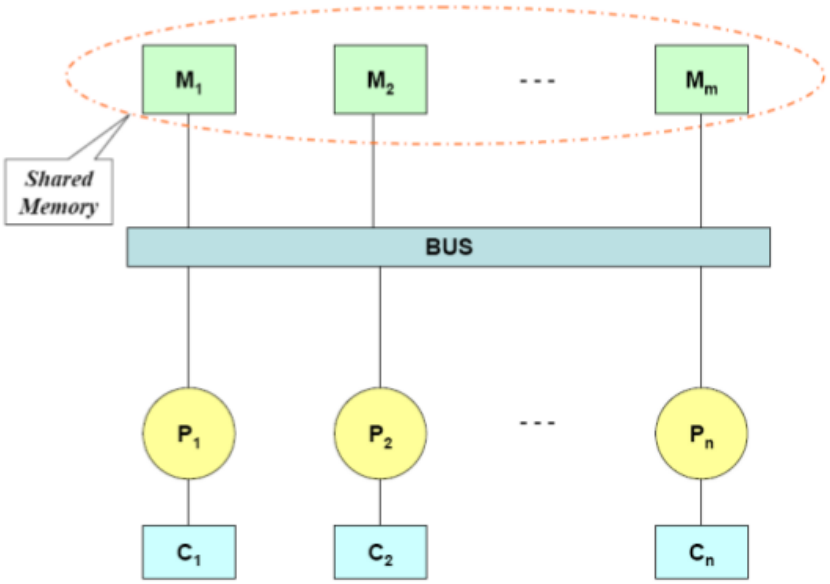


Figura 54: Arquitectura MIMD de tipus Memòria Compartida (UMA, SMP)

3. **Memòria Distribuïda i Compartida**

- Són arquitectures on es connecten processadors de memòria distribuïda mitjançant un bus i construint per sobre una arquitectura de memòria compartida.
- Un processador pot accedir al seu mòdul de memòria directament sense passar pel bus i pot accedir als mòduls dels altres mitjançant aquest bus.
- Degut a que els processadors accedeixen amb diferents latències a diferents mòduls de memòria se'ls anomena Non Uniform Memory Access (NUMA).

- L'accés a mòduls de memòria externa es fa de la mateixa forma que en processadors de memòria compartida, mitjançant variables compartides.
- S'utilitza en màquines com per exemple els SGI, el Marenostrum, els ALTIX, BlueGene, i avui dia en tota màquina amb més d'un processador físic amb diversos nuclis a la placa base.
- En realitat es fa servir la variant ccNUMA. Aquesta tecnologia pren el nom per “cache coherent NUMA” i soluciona el problema de que quan diversos processadors volen accedir a la memòria molt ràpidament, les caches de tots els altres han de quedar invalidades. S'utilitzen algorismes tant des de el hardware com des de el SO per accelerar la coherència de totes les caches de tots els processadors. Exemples de protocols utilitzats són MESIF o SCI [42]. El Xeon E5645 fa servir el protocol MESIF.

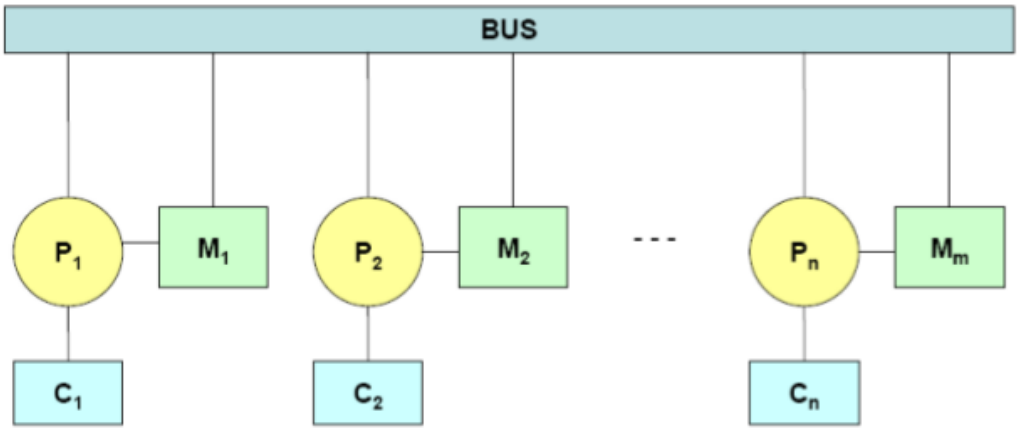


Figura 55: Arquitectura MIMD de memòria distribuïda i compartida (DSM o NUMA)

4.2.2.2 Arquitectura dels servidors i el clúster de CIMNE

Per identificar els tipus de sistemes dels que disposem hem de veure una mica la història dels processadors Intel i AMD.

Intel fins a finals de 2008 distribuïa processadors que feien ús del que s'anomena el Front Side Bus (FSB). El Front Side Bus era el circuit principal que comunicava els processadors amb el chipset. El chipset disposava del north bridge i del south bridge, el primer era bàsicament el controlador de memòria principal i el segon el qui controlava l'E/S dels diferents dispositius i busos de la placa base. En un esquema com aquest es tenia una memòria principal que era compartida amb els dos processadors mitjançant el controlador de memòria i per tant es disposava d'un esquema UMA.

A partir de Març de 2009 Intel va incorporar en els seus processadors Xeon i Nehalem el que s'anomena el Quickpath Interconnect (QPI), [37]. Aquesta tecnologia és la vigent actualment i consisteix en incorporar el controlador de memòria dins el propi processador assignant a cada processador uns bancs de memòria RAM determinats. D'aquesta manera es creava un esquema NUMA permetent que un processador tingués un accés local a memòria RAM molt ràpid i per altra banda que també tingués accés a la memòria de l'altre processador mitjançant el QPI. Habitualment aquesta característica es pot deshabilitar des de les BIOS amb el paràmetre Node Interleaving que permet convertir l'arquitectura NUMA [39] amb una UMA o SMP [40].

No obstant, Intel no va ser l'inventor de la tecnologia QPI. AMD portava des de 2001 fabricant els seus processadors Opteron, Athlon, Turion, Sempron, etc. amb tecnologia HyperTransport. Aquesta tecnologia és la mateixa que va copiar Intel el 2009, amb la diferència que és un

estàndard obert i lliure promogut per l'HyperTransport Consortium [43]. Anteriorment molts altres processadors com els de Nvidia, PowerPC, ATI, SiByte MIPS de Broadcom, etc. ja utilitzaven l'HT.

Per altra banda hem de comentar que des de la introducció dels processadors multi-nucli al voltant de l'any 2004 es va combinar la tecnologia SMP i la NUMA amb el que hem comentat en l'apartat anterior que s'anomenava com a arquitectura DSM, Distributed Shared Memory. Aquest fet va fer que els processadors tinguessin assignats uns certs mòduls de memòria, i cada un del seus nuclis compartís aquests de forma uniforme tal com ho fa l'arquitectura SMP o UMA.

Amb aquest breu repàs a la cronologia dels processadors ens trobem amb condicions de determinar de quines arquitectures disposem a CIMNE.

Primer de tot determinem que els processadors que tenim són AMD Opteron per el Sun Fire x4600, el Sun Ultra 40 i els nodes del clúster M605. Per altra banda els nodes M600 disposen de Intel Xeon E5410 amb data de sortida de novembre de 2007, microarquitectura Harpertown. Finalment els nodes M610 són nodes amb Intel Xeon E5645 amb microarquitectura Westmere-EP, successora de la Nehalem.

Podem deduir per tant que totes les arquitectures de les que disposem, excepte la dels nodes M600, utilitzen la forma d'accés a memòria NUMA i un model DSM ja que tots els processadors són multi-nucli. Els nodes M600 disposen de l'antic FSB i per tant fan servir l'accés UMA o SMP exclusivament.

Al disposar d'aquestes arquitectures, en tots els casos excepte en els M600 ens interessa disposar d'eines al sistema operatiu capaces de determinar la millor configuració d'allotjament de processos als nuclis del processador en relació a les posicions de memòria utilitzades. Per fer-ho el kernel ha de ser compilat amb la opció CONFIG_NUMA i llavors podem fer servir la llibreria libnuma o l'eina numactl com recomanen al *man-page(7)* de Linux. Aquesta permet entre altres coses determinar les millors opcions d'afinitat de processos, permet veure, amb la comanda numastat l'estadística de falls i accessos en els diferents processadors, etc.

Les diferents sortides de numastat confirmen que en tots els casos disposem d'arquitectura NUMA excepte en els M600:

Al fer un numastat en un node M600 (2x Intel Xeon E5410 amb FSB) obtenim que tota la memòria és local al domini de NUMA anomenat node0:

```
[root@pez006 ~]# numastat
                           node0
numa_hit              2237728360
numa_miss                    0
numa_foreign                 0
interleave_hit           20682
local_node            2237728360
other_node                   0
```

Si ho feim a un M610 (2x Intel Xeon E5645 amb QPI) veim dos dominis NUMA, node0 i node1, i a més observem com no tots els accessos realitzats són locals:

```
[root@pez014 ~]# numastat
                           node0           node1
numa_hit                55049419        51523737
numa_miss                1228958          227001
numa_foreign              227001         1228958
interleave_hit             19621           19624
local_node              55048907        51500909
other_node               1229470          249829
```

Si ho feim a un M605 (2x AMD Opteron 2356 amb HT) obtenim el mateix resultat que a l'M610:

```
[root@pez012 ~]# numastat
                           node0           node1
numa_hit                426875513       388555805
numa_miss                       0        27240153
numa_foreign             27240153               0
interleave_hit              13345           13418
local_node              426873371       388539972
other_node                   2142        27255986
```

Si ho feim al node Fire X4600 (4x AMD Opteron 880 amb HT):

```
[root@XFire ~]# numastat
                           node0           node1           node2           node3
numa_hit                  100628           91668           97534          136546
numa_miss                      0               0               0               0
numa_foreign                   0               0               0               0
interleave_hit             14602           14601           14597           14601
local_node                100114           77052           82925          122093
other_node                   514           14616           14609           14453
```

A més de l'eina numactl del sistema operatiu, la tasca del kernel és centrarà en gestionar les capacitats de Symmetric Multi Processing (SMP) de cada processador. D'aquesta manera el kernel haurà de ser compilat per poder utilitzar diversos processadors i nuclis amb el paràmetre CONFIG_SMP=y. Aquesta opció haurà de ser activada en tots els servidors dels que disposem. Per comprovar que el kernel suporta SMP ho fem amb la comanda “uname -a”:

```
[root@acuario ~]# uname -a
Linux acuario 2.6.32-220.23.1.el6.x86_64 #1 SMP Mon Jun 18 09:58:09
CDT 2012 x86_64 x86_64 x86_64 GNU/Linux
```

Com a referència il·lustrativa a la Figura 56 veiem un esquema representatiu de dos processadors de quatre nuclis connectats mitjançant la tecnologia QPI i tenint cadascun d'ells tres canals d'accés a bancs de memòria individuals [38].

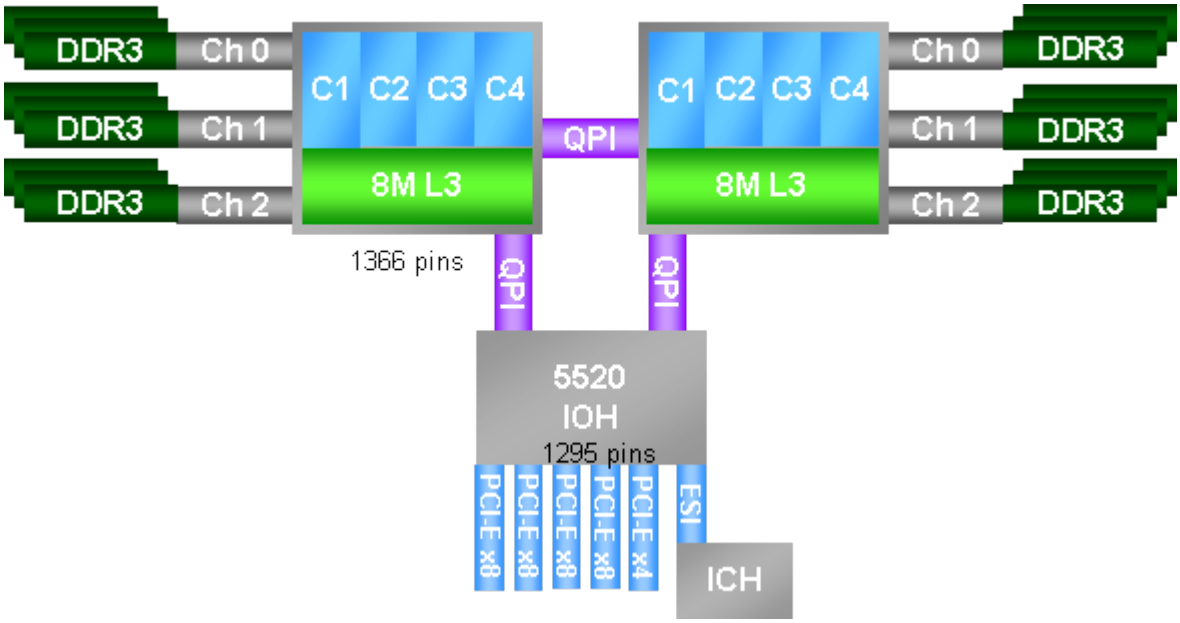


Figura 56: Arquitectura NUMA amb connexions QPI i SMP dins cada processador

Per acabar l'apartat hem de dir que en el cas que ens ocupa estem configurant un clúster d'HPC amb diversos nodes on l'arquitectura entre nodes és l'equivalent a una arquitectura distribuïda: un processador d'un node X no pot accedir directament a la memòria d'un processador d'un node Y.

D'aquesta manera el que tindrem és un sistema que tindrà l'arquitectura d'un clúster de memòria distribuïda on per comunicar un procés d'un node amb un procés d'un altre s'haurà de disposar d'una xarxa pel pas de missatges. Aquesta xarxa serà la Infiniband i juntament amb les biblioteques de programació adequades (p.ex. MPI) simularà el que seria la memòria compartida entre diferents nodes.

4.2.3 El clúster de memòria distribuïda

En aquest apartat investiguem quines són les alternatives que existeixen actualment per construir un clúster que utilitza un esquema de memòria distribuïda entre nodes. Serà pràcticament idèntic a muntar un clúster Beowulf (tot i que amb hardware específic) i per tant serà vàlid també el cercar solucions d'aquest entorn.

Primer de tot veurem quines parts composen un clúster d'aquestes característiques i a continuació veurem les alternatives actuals.

4.2.3.1 Arquitectura del sistema

Hem confeccionat un esquema que organitza el sistema amb 4 capes segons la funcionalitat que aporten, les descrivim a continuació.

Capa 1. Sistema Operatiu
Aquesta capa té com a funcionalitat proporcionar la base per totes les altres capes i és la de més baix nivell. Realitza la comunicació amb el hardware i proporciona interfícies d'accés als recursos del sistema, xarxes, etc.

Capa 2. Seguretat i monitoreig
Aquesta capa és comuna a tot servidor, ja sigui de HPC o de qualsevol altre tipus. Proporciona seguretat al sistema i eines per controlar-ne els recursos disponibles i gestionar-los adequadament. També determina les polítiques d'accés, implementa sistemes anti-atacs i proporciona informació de l'estat del hardware amb diverses eines.

Capa 3. Serveis de clustering
Aquesta és la capa més important d'un clúster HPC i disposa de tots els serveis que converteixen un servidor amb un clúster. La part més important és el gestor de recursos i planificador de treballs, que acaba sent qui controla i manega finalment els treballs dels usuaris. Per tal que aquest sistema funcioni calen altres subsistemes de la mateixa capa tals com la sincronització entre software dels nodes, un sistema de desplegament d'imatges, sincronisme amb els usuaris del cluster a tots els nodes, etc.

Capa 4. Interfícies d'administració & usuari
Finalment aquesta capa permet controlar el gestor de recursos i planificador de la capa anterior, a més de permetre controlar el clúster en global, el seu software, actualitzacions, etc. Tot això per part de l'administrador. Per part de l'usuari proporciona el software i biblioteques necessàries perquè el científic pugui realitzar els seus càlculs. Com és lògic la capa 3 i la capa 4 tenen eines comunes i es recolzen mutuament.

Veiem a la Figura 57 l'estructura en capes que hem confeccionat i que resumeix tots els components necessaris per tal d'instal·lar un clúster de memòria distribuïda.

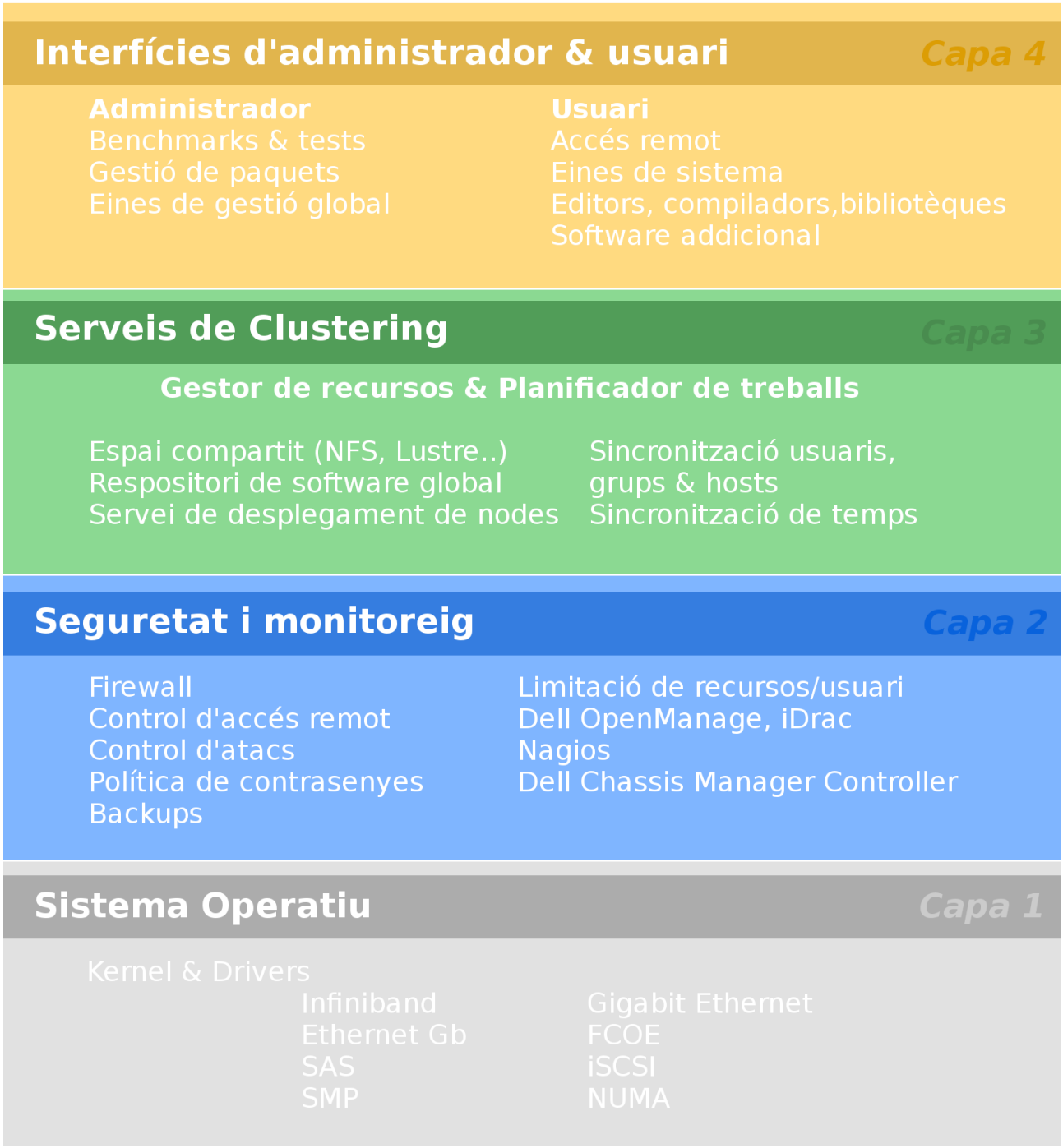


Figura 57: Estructura d'un clúster de memòria distribuïda

4.2.4 Gestors de recursos i planificadors de treballs

El gestor de recursos i el planificador de treballs són la part més important que defineix un clúster de computació. Cada un d'aquests gestors requereix realitzar una configuració determinada per les capes 3 i 4 explicades anteriorment, i és per això que l'elecció d'aquest condicionarà i determinarà com implementar la resta del sistema.

SLURM

De Simple Linux Utility for Resource Manager es tracta d'un gestor de recursos desenvolupat per el grup SchedMD LLC *http://www.schedmd.com/* i patrocinat per la NNSA, National Nuclear Security Administration *http://www.nnsa.energy.gov* .

Disseny i programació:

- Escrit en C
- Modular amb possibilitat d'afegir plugins
- Paradigma Client – Servidor
- Tolerància a falls amb servidors secundaris

Característiques:

- Control de recursos per usuaris i grups
  - Multi-factor
  - Prioritats
  - QoS
  - Crèdits
- Planificador integrat
  - FIFO
  - Backfill
  - Gang Scheduling
  - Preemption
- Possibilitat d'afegir planificador extern
- Fins a 120.000 treballs per hora amb més de 2 milions de processadors
- Apte per clústers heterogenis
- Afinitat de tasques a nivell de nucli de CPU, socket o node.
- Suport per la majoria d'implementacions d'MPI
- Particions de nodes amb propietats (cues)
- Recuperació de treballs (checkpoints)
- Funcions d'estalvi d'energia
- Seguretat OpenSSL i autenticació
- Possibilitat de càlcul GPU

Grau de dificultat:	Fàcil
Interfície gràfica:	Si
Plataformes:	AIX, Linux, OS X, Solaris
Xarxes:	BlueGene, Cray XT i XE, Ethernet, IBM Federation, Infiniband, Myrinet, Quadrics Elan, Sun Constellation.
Documentació:	Extensa, actualitzada i clara.
Suport:	Opció de pagament amb diversos serveis i comunitat de desenvolupadors i usuaris activa.
Llicència i preu:	OpenSource, gratuït.

Exemples d'utilització:

- *Tianhe-1A* designed by *The National University of Defence Technology (NUDT)* in China with 14,336 Intel CPUs and 7,168 NVDIA Tesla M2050 GPUs, with a peak performance of 2.507 Petaflops.
- *Tera 100* at *CEA* with 140,000 Intel Xeon 7500 processing cores, 300TB of central memory and a theoretical computing power of 1.25 Petaflops. Europe's most powerful supercomputer.
- *LOEWE-CSC*, a combined CPU-GPU Linux cluster at *The Center for Scientific Computing (CSC)* of the Goethe University Frankfurt, Germany, with 20,928 AMD Magny-Cours CPU cores (176 Teraflops peak performance) plus 778 ATI Radeon 5870 GPUs (2.1 Petaflops peak performance single precision and 599 Teraflops double precision) and QDR Infiniband interconnect.
- *Dawn*, a BlueGene/P system at LLNL with 147,456 PowerPC 450 cores with a peak performance of 0.5 Petaflops.
- *Rosa*, a CRAY XT5 at the *Swiss National Supercomputer Centre* named after Monte Rosa in the Swiss-Italian Alps, elevation 4,634m. 3,688 AMD hexa-core Opteron @ 2.4 GHz, 28.8 TB DDR2 RAM, 290 TB Disk, 9.6 GB/s interconnect bandwidth (Seastar).
- *EKA* at Computational Research Laboratories, India with 14,240 Xeon processors and Infiniband interconnect
- ***MareNostrum* a Linux cluster at the *Barcelona Supercomputer Center* with 10,240 PowerPC processors and a Myrinet switch**
- *Anton* a massively parallel supercomputer designed and built by *D. E. Shaw Research* for molecular dynamics simulation using 512 custom-designed ASICs and a three-dimensional torus interconnect.

PBS

De Portable Batch System, és un software planificador de tasques que fou desenvolupat a l'any 1991 per la NASA a càrrec de MRJ Technology Solutions (Veridian). Finalment Altair Engineering va adquirir el software i la porpietat intel·lectual de Veridian l'any 2003. El 1998 sorgí una versió gratuita de codi lliure anomenada OpenPBS que no està actualment en funcionament.

Disseny i programació:

- Paradigma Client – Servidor

Característiques:

- Càlcul en GPU
- Planificador integrat, ajustable
  - Backfill
- Canvi de sistema operatiu en funció del treball a llançar
- Augment de recursos del treball durant el funcionament
- Apte per configuracions amb molts nodes
- Polítiques d'usuari i grup
- Gestor de recursos
- Ús d'scripts python
- Suport per implementacions MPI i OpenMP

Grau de dificultat:	Normal
Interfície gràfica:	Si
Plataformes:	Windows, RedHat, OS X, SGI, Oracle, Suse - IBM, Dell, Acer, Bull, Cray, HP
Xarxes:	Infiniband, SGI, Cray, IBM i GigE
Documentació:	Completa i actualitzada, disponible al web *http://www.pbsworks.com*
Suport:	Opció de pagament amb diversos serveis, comunitat on-line.
Llicència i preu:	De pagament, 14.25€/core / any (aprox. 1.995€/any), codi propietari.

Exemples d'utilització:

- Chrysler - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=351&*
- Univeristy of Florida - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=50&*
- TRW Automotive - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=51&*
- Boeing - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=48&*
- GE's Oil & Gas business - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=57&*
- Ford - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=59&*
- NASA - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=55&*
- LSU - *http://www.pbsworks.com/ResLibDownload.aspx?file_id=56&*

TORQUE Resource Manager

Terascale Open-Source Resource and Queue Manager, és com el seu nom indica un gestor de recursos i planificador de cues. Els desenvolupadors són el grup Adaptative Computing *http://www.adaptivecomputing.com/* i és un software que porta funcionant des de l'any 2003.

Torque sorgeix del projecte PBS gràcies a un esforç de la comunitat. Organitzacions com la NCSA, OSC, USC, TeraGrid, etc. han col·laborat amb el seu desenvolupament. Habitualment Torque s'integra amb el Maui Cluster Scheduler o el Moab Workload Manager per millorar la planificació de treballs i administració d'aquest.

Programació:

- Escrit en C

Característiques:

- Tolerància a falls en nodes
- Millores en el planificador respecte a PBS
- Millores d'escalabilitat en quant a processadors i nombre de tasques
- Més monitorització del sistema

Grau de dificultat:	Difícil
Interfície gràfica:	No
Plataformes:	Sistemes UNIX
Xarxes:	Infiniband, SGI, Cray, IBM i GigE
Documentació:	Al moment d'aquest treball, deficient e incompleta
Suport:	Opció de pagament a Adaptative Computing, comunitat parcialment inactiva
Llicència i preu:	Codi no lliure però obert, llicència pròpia. Gratuït.

Exemples d'utilització:

- *University of Cambridge: Accounting and Cost Recovery*
- *Barcelona Supercomputing Center*
- *TriLabs (Department of Energy)*
- *Oak Ridge National Laboratory (Jaguar)*
- *The Weather Channel*
- *Penguin Computing Public HPC Cloud*
- *University of Birmingham - BlueBear system*
- *CHPC South Africa*
- *University of Cambridge: Multi-OS Management*
- *Rocky Mountain Supercomputing*

Sun Grid Engine – Oracle Grid Engine

En el moment de la realització d'aquest projecte la companyia SUN acabava de ser comprada per Oracle. En aquest moment el futur de SGE, el sistema gestor de treballs i recursos per excel·lència de SUN estava compromès. No es tenia clar com Oracle el seguiria desenvolupant i tot apuntava a que faria el codi que fins aleshores era lliure amb codi propietari. Més endavant succeí que SGE canvià de nom a OGE, Oracle Grid Engine i que en versions posteriors a la 6.2u7 el codi seria comercial.

Característiques:

- Control basat en regles i quotes
- Execució remota sense sshd
- Multi-clúster
- Treballs interactius
- Interfície gràfica avançada per seguiment de processos
- Integració amb Hadoop i Amazon EC2
- Diversos algorismes de planificació
- Cues de treballs
- Tolerancia a falls amb múltiples servidors
- Recuperació de treballs (checkpoints)
- Conjunts de jobs i tasques
- Suport per MPI, PVM, OpenMP
- Control d'ús
- Compiladors paral·lels de SUN

Grau de dificultat:	Normal
Interfície gràfica:	Si
Plataformes:	AIX, BSD, HP-UX, IRIX, Linux, OS X, Solaris, Super-UX, Tru64, Windows (només nodes), Z/OS
Xarxes:	Infiniband, SGI, Cray, IBM, GigE, etc.
Documentació:	Completa i actualitzada, disponible al web *http://www.oracle.com*
Suport:	Opció de pagament amb diversos serveis
Llicència i preu:	De pagament, 383€/processador + 1y support (aprox. 14.921€) codi propietari.

Exemples d'utilització:

- Mentor Graphics – *http://www.mentor.com*
- Complete Genomics – *http://www.completegenomics.com/*
- Rising Sun Pictures - *http://www.rsp.com.au/*
- Texas Advanced Computing Center at the University of Texas - *http://www.tacc.utexas.edu/*

IBM® Platform™ LSF®

Load Sharing Facility és un software comercial planificador de tasques. El 2007 la casa Platform Computing va alliberar una versió simplificada de LSF anomenada Platform Lava (*http://www.openlava.org/*) amb llicència GPLv2. La última versió estable és de Gener de 2011. Posteriorment la companyia fou comprada per IBM i el software ara s'anomena IBM® Platform™ LSF®.

(*http://www-03.ibm.com/systems/technicalcomputing/platformcomputing/products/index.html*)

Característiques:

- Plataformes heterogènies
- Algorismes que detecten la millor distribució de treball possible
- Capacitat per ajustar-se a prioritats del negoci
- Canvis sense tenir que re-iniciar els serveis de cluster
- Planificador de tasques
  - Fair share
  - Bulk job
  - Preemption
  - Backfill
  - Reserves
  - Control de “job starving”
  - Recuperació de treballs (checkpoint)
  - Càlcul amb GPU
  - Control d'energia

Grau de dificultat:	Normal
Interfície gràfica:	Si
Plataformes:	Linux
Xarxes:	Infiniband, IBM, GigE, etc.
Documentació:	Completa i actualitzada
Suport:	Opció de pagament amb diversos serveis (IBM), lliure i comunitat OpenLava.
Llicència i preu:	De pagament i propietari (IBM) o GPLv2 (OpenLava).

Exemples d'utilització:

- Red BullRacing - *http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=AB&infotype=PM&appname=STGE_DC_ZQ_USEN&htmlfid=DCC03022USEN&attachment=DCC03022USEN.PDF*
- A*STAR - *http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=AB&infotype=PM&appname=STGE_DC_ZQ_USEN&htmlfid=DCC03025USEN&attachment=DCC03025USEN.PDF*

MOAB

Moab Cluster Suite és un software desenvolupat per Adaptative Computing, mateixos desenvolupadors de Torque i que pretén ser un conjunt d'eines que compten amb un planificador de tasques avançat, un sistema d'administració gràfica per el clúster i un portal per l'usuari per controlar, enviar i gestionar els seus treballs.

MOAB funciona en conjunt amb un altre gestor de recursos que el suporti, com pot ser TORQUE, SLURM, PBS, etc.

A la Figura 58 veiem un exemple de la pila de protocols de MOAB, on les parts d'aquest es troben anomenades com a Batch Workload i Cluster Workload.

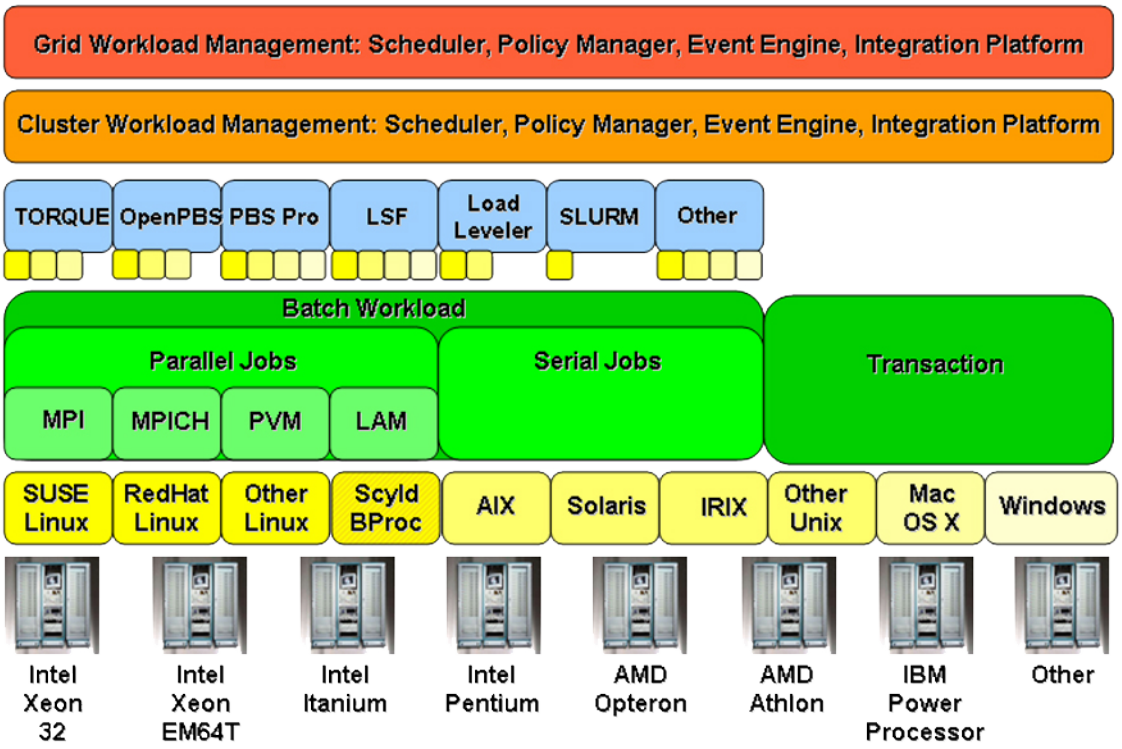


Figura 58: Esquema del MOAB stack

MOAB fa servir planificadors avançats per gestionar, controlar i informar de les càrregues del clúster en massa i en instal·lacions heterogènies. Disposa d'algorismes de planificació patentats amb un motor intel·ligent que utilitza polítiques multi-dimensionals per accelerar càrregues de forma ideal sobre diversos recursos.

Exemples d'institucions que ho utilitzen es poden trobar a la mateixa llista que els que fan servir TORQUE.

MOAB no és gratuït i es demana registre per tal de obtenir un pressupost. És una bona opció i el grup de Adaptative Computing ofereix diversos paquets i serveis, per més informació es pot visitar la seva pàgina web: *http://www.adaptivecomputing.com/*

MAUI

Es tracta de la implementació predecessora de MOAB, OpenSource i gratuïta. Disposa de moltes característiques similars a MOAB però sovint està menys actualitzada i no ofereix eines gràfiques i les facilitats d'administració de MOAB.

Les diferències concretes que MAUI no té en front a MOAB són: clusters privats virtuals, suport per triggers, eines d'administració gràfica i portal web per els usuaris.

És gratuït i funciona sobre PBS, Loadleveler, SGE, BProc, SSS XML, LSF o usant una API de planificador anomenada Wiki FlatText.

http://www.clusterresources.com/products/maui-cluster-scheduler.php

4.2.5 Paràmetres d'un planificador de treballs

Hem analitzat les opcions de configuració d'un planificador de treballs, concretament SLURM, per veure quines possibilitats tenim. En general serà semblant a tot altre gestor de recursos.

A partir d'aquí, s'entén per *"job"* un treball enviat per un usuari al planificador, ja sigui sèrie o requereixi diversos fils per córrer.

Job Priority

Quina prioritat donem als jobs?

Opció 1: Bàsic, assigna la prioritat en funció de l'arribada

Opció 2: Multi-factor

Prioritat en funció de:

- Age: Quant de temps fa que el job està esperant
- Fair-share: L'usuari ha calculat molt anteriorment?
- Partition: Partició (o cua) a la que va dirigida el Job
- QoS: Factor que defineix l'usuari, segons la urgència que té

Job Scheduler

Quan executem els jobs?

Opció 1: Builtin, esquema first-in-first-out

Opció 2: Backfill, inicia una tasca de baixa prioritat si fer-ho no suposa retardar l'inici d'una tasca de més prioritat. Essencialment, jobs petits omplen forats. Aquesta opció va lligada a especificar temps estimats pels jobs.

Opció 3: Gang, permet parar i iniciar diferents jobs mentre estan corrent intentant maximizar el temps utilitzat i els recursos utilitzats. Alguns jobs poden compartir recursos, i hi haurà pauses en les execucions però així un job petit no haurà d'esperar un de gran perquè comenci. La seva funció és maximitzar l'ús del clúster.

Exemple:

```
Job 1 usa 2 nodes.
Job 2 espera per 3 nodes.
Job 3 espera per 1 node.
```

Llavors el Job 3 en condicions normals hauria d'esperar a que el Job 2 acabés i no podría passar davant.

Gang permet que si el temps restant per Job 1 és inferior al temps d'execució de Job 3, com que Job 3 no interferirà a Job 3, pugui entrar a calcular.

Opció 4: Preemption, un job d'altra prioritat expulsa un job de baixa prioritat, llavors aquest es cancela, es suspen, o s'encua de nou (possiblement a altres particions).

User Accounting

Aquesta funció permet controlar quines prioritats es donen als usuaris, defineix a quines particions poden córrer, quants jobs al mateix temps, quins límits de temps i recursos, etc. Ho fa basant-se amb paràmetres individuals o de grup.

Particions

Conjunts de nodes amb propietats com: nombre de treballs concurrents màxims, temps màxim de càlcul per treball, recursos compartibles o no, prioritat pre-definida, etc. En un altre gestor de recursos també es poden anomenar “cues”.

4.2.6 Solucions actuals

A la vista de la investigació realitzada les solucions actuals es concentren habitualment en dos tipus:

1. Solució gratuïta i OpenSource

   En aquest cas s'utilitza software OpenSource que es pot implementar de forma manual o en forma de paquet preparat.

   - Mètode manual: Consisteix en implementar manualment totes les capes que hem comentat a l'apartat 4.2.3.1 Arquitectura del sistema.

   - Paquet preparat: S'utilitza un software que s'instal·la a sobre de la capa 2 (Sistema operatiu + Seguretat i monitoreig) i implementa totes o gairebé totes les funcionalitats de la capa 3 i 4.

2. Solució comercial

   Aquesta solució comporta la contractació d'uns serveis a una empresa dedicada al HPC i que realitza una instal·lació de totes les capes del sistema. Opcionalment inclou software de gestió, control i interfícies d'usuari molt atractives per l'administrador i els usuaris.

A mode d'exemple comentem algunes solucions gratuïtes en format de paquet preparat que s'han valorat en el moment de fer el projecte:

- OSCAR: Paquet preparat que implementa tot el necessari per disposar d'un clúster de tipus Beowulf o memòria distribuïda a sobre d'un sistema operatiu basat en RedHat Enterprise 5 o Debian 6. La última versió és la 6.1.1 de Maig de 2011.

  La documentació del projecte OSCAR és molt pobre només disposant de dos documents poc revisats. La comunitat també està inactiva i de fet a data de 09/2012 la versió del software no ha estat actualitzada.

  Aquest paquet de software és el que va instal·lar LINALCO a CIMNE el 2008. Es pot obtenir més informació de OSCAR a la pàgina web oficial:

  http://svn.oscar.openclustergroup.org/trac/oscar

- ROCKS: De *http://www.rocksclusters.org* aquest software s'ofereix en dues categories. Una inclou tot el sistema operatiu + paquest, i l'altre només el “toolkit” que prepara l'entorn per un cluster. Al moment d'estudi d'aquest projecte la versió actual era la Rocks 5.3 basada en Centos 5.3, essent Centos 6.0 la més actual.

  La documentació és bona i a més és un sistema actualitzat i utilitzat de forma notable.

- CHAOS i OpenMosix: Aquesta distribució de Linux estava dissenyada per crear xarxes ad hoc de clústers. Ocupava 6MB i al ser instal·lada proporcionava un entorn openMosix, un sistema complet de clúster amb distribució de càrrega entre nodes, treballs paral·lels, etc. La última versió estable fou la 1.6 d'Agost de 2005. *http://midnightcode.org/projects/chaos/*

- Caos NSA & Perceus: Caos Linux és una distribució de Linux orientada tant a servidors, com a clústers o usuaris finals, tal com indica l'acrònim NSA; Node, Server, Appliance. Les seves principals característiques és que és basada en un sistema de paquets RPM, ofereix actualitzacions per més de 3 anys, és gratuïta i OpenSource i s'enfoca cap a la computació d'alt rendiment.

  S'utilitza en clústers de fins a 50.000 nodes és popularment estable i ràpida. És un projecte de Infiscale *http://www.infiscale.com/* .

Disposa de suport comercial per part de Infiniscale, llistes de correu, Wiki, suport per xat, etc.

Perceus és un paquet de software que permet realitzar el desplegament de imatges i sistemes operatius en un entorn de clúster. La combinació Caos NSA & Perceus s'està fent popular i és recomanada per els desenvolupadors de SLURM.

- GOLD és un software desenvolupat per el Pacific Northwest National Laboratory que permet fer un seguiment acurat sobre els recursos utilitzats en un clúster HPC. A partir d'aquest seguiment actua com un banc i estableix comptes amb crèdit que se li assignen als usuaris. Els usuaris poden fer servir els recursos del clúster pagant amb aquests crèdits i per tant obtenint un control de la utilització. Es pot integrar amb MOAB.

  http://www.clusterresources.com/products/gold-allocation-manager.php

Com a solucions comercials disposem de moltes companyies que ens oferirien la instal·lació que necessitem amb diferents prestacions i sistemes. Ja hem comentat les solucions de IBM i Oracle, però també hi hauria disponibles altres com les de Bull *www.**bull**.es* o Fujitsu

http://www.fujitsu.com/global/services/solutions/tc/hpc/ .

Finalment cal fer una menció als serveis actuals de computació al núvol per exemple de part de Amazon EC2 o Google Compute.

Aquests ofereixen capacitat de càlcul en remot amb el nombre de nodes que vulguem i disponibles de forma dinàmica. També ofereixen solucions de computació amb GPUs.

Descartarem aquestes opcions ja que volem aprofitar els equipaments actuals. Per més informació es poden visitar les seves pàgines web:

Amazon EC2 - *http://aws.amazon.com/ec2/pricing/*

Google Compute Engine - *http://cloud.google.com/products/compute-engine.html*

4.3 Models de programació paral·lela

Com sabem la paral·lelització d'un codi suposa dividir la part de major pes d'aquest codi en fragments més petits amb l'objectiu de distribuir-los a altres unitats computacionals, seguint sempre la llei d'Amdahl. Aquestes unitats poden ser nuclis d'un processador o en altres ocasions computadors diferents, o una combinació d'ambdós.

Pel cas que ens ocupa hem de diferenciar els dos tipus de paral·lelització. El primer cas consisteix en la que un codi es divideix en diferents fils d'execució mitjançant crides al sistema i deixa al kernel que s'encarregui de distribuir aquests fils amb els processadors o nuclis disponibles a la màquina i el segon cas consisteix en executar uns agents en un node diferent i comunicar-s'hi mitjançant el pas de missatges enviant-li treballs.

Sovint els dos models de paral·lelització es combinen.

Sobre el tema de la parel·lització i els diferents models de programació podem trobar una guia molt extensa i que recomanem llegir en profunditat a la pàgina de LLNL [55].

4.3.1 Programació amb múltiples fils d'execució

Aquest model de programació és un tipus de programació que utilitza memòria compartida.

Una possible analogia que podem fer per descriure el model de programació, és pensar en un programa que conté diverses subrutines.

Quan aquest procés és executat, diem-li a.out, el sistema operatiu li assigna una sèrie de recursos com un identificador (PID), una determinada quantitat de memòria, una prioritat, etc. Llavors a.out realitza una sèrie de treballs un darrera l'altre (treball en sèrie), i després crea un conjunt de tasques (fils d'execució) que poden ser programades per el kernel concurrentment.

Cada fil d'execució disposa de dades locals no accessibles pel pare ni pels altres fills, però sí que té accés als recursos del seu pare i al seu espai d'adreces. El fet de que comparteixi aquests recursos impedeix que se n'hagin de tornar a assignar per cada fil creat.

Una vegada creats aquests fils d'execució, el kernel els planifica potser en diferents nuclis d'un mateix processador i tots treballen al mateix temps. Entre ells es comuniquen mitjançant la memòria global. D'aquesta manera es requereix un mecanisme de sincronització per assegurar que no dos fils no creen dependències de lectura-escriptura o escriptura-escriptura. Figura 59.

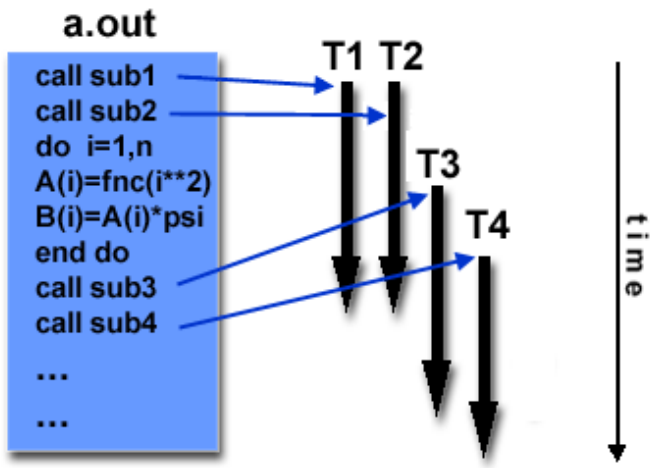


Figura 59: Model de programació en fils d'execució

Des del punt de vista del programador, per realitzar un programa que utilitzi aquest model de programació es necessiten una d'aquestes dues coses:

- Conjunt de biblioteques cridades des del codi paral·lel.
- Conjunt de directives dirigides al compilador, en el codi sèrie o paral·lel.

En ambdós casos, el programador és el responsable de programar el paral·lelisme.

Com a nota important, els processadors amb “hyper-threading” implementen una sèrie de millores per accelerar el canvi de context entre diferents fils d'execució.

S'han realitzat moltes implementacions de tots dos tipus, però al final s'han consolidat en dues molt diferents: POSIX Threads i OpenMP.

a) Pthreads

- Basats en biblioteques, requereix escriure codi paral·lel
- Especificad per l'estandard IEEE POSIX 1003.1c (1995).
- Només llenguatge C
- Tutorial de LLNL: *https://computing.llnl.gov/tutorials/pthreads/*

b) OpenMP

- Basat en directives del compilador. Pot fer servir codi sèrie.
- Definit per un grup de fabricants de hardware i software, la API per Fortran és de 1997, la de C/C++ de 1998.
- Portable i multi-plataforma.
- C, C++ i Fortran.
- Senzill d'utilitzar
- Tutorial de LLNL: *https://computing.llnl.gov/tutorials/openMP/*

Hello World! amb Pthreads

```
#include <pthread.h>
#include <stdio.h>
#define NUM_THREADS     5

void *PrintHello(void *threadid)
{
   long tid;
   tid = (long)threadid;
   printf("Hello World! It's me, thread #%ld!\n", tid);
   pthread_exit(NULL);
}

int main (int argc, char *argv[])
{
   pthread_t threads[NUM_THREADS];
   int rc;
   long t;
   for(t=0; t<NUM_THREADS; t++){
      printf("In main: creating thread %ld\n", t);
      rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
      if (rc){
           printf("ERROR; return code from pthread_create() is %d\n",
rc);
         exit(-1);
      }
   }

   /* Last thing that main() should do */
   pthread_exit(NULL);
}
```

Resultat:

```
[user@acuario]$ ./hello
In main: creating thread 0
In main: creating thread 1
Hello World! It's me, thread #0!
In main: creating thread 2
Hello World! It's me, thread #1!
In main: creating thread 3
Hello World! It's me, thread #2!
In main: creating thread 4
Hello World! It's me, thread #3!
Hello World! It's me, thread #4!
```

Hello World! en OpenMP

```
#include <omp.h>

main ()  {

int nthreads, tid;

/*  Fork  a  team  of  threads  with  each  thread  having  a  private  tid
variable */
#pragma omp parallel private(tid)
  {

  /* Obtain and print thread id */
  tid = omp_get_thread_num();
  printf("Hello World from thread = %d\n", tid);

  /* Only master thread does this */
  if (tid == 0)
    {
    nthreads = omp_get_num_threads();
    printf("Number of threads = %d\n", nthreads);
    }

  }  /* All threads join master thread and terminate */

}
```

Resultat:

```
[user@acuario]$ ./hello
Hello World from thread = 0
Number of threads = 2
Hello World from thread = 1
```

4.3.2 Programació per pas de missatges

Aquest model de programació és un tipus de programació que utilitza memòria distribuïda.

Existeix un conjunt de tasques que fan servir la seva pròpia memòria durant l'execució. Les tasques poden trobar-se en la mateixa màquina física o en màquines diferents, o les dues coses al mateix temps. Figura 60.

Les tasques s'intercanvien la informació utilitzant comunicacions per la xarxa, enviant-se missatges. Aquestes transferències d'informació normalment necessiten ser recolzades per operacions compartides que han de realitzar tots els processos, per exemple una operació d'enviar dades de A a B requereix que a A hi hagi l'operació d'enviar i que B esperi aquestes dades.

La implementació d'aquest model de programació es fa mitjançant crides a biblioteques. Aquestes crides es trobaran dins el codi font i per tant el programador és el responsable de determinar tot el paral·lelisme.

El 1992 es va crear l'MPI Forum [56] amb l'objectiu de establir un estàndard per les diferents implementacions que existien en el moment del pas de missatges entre tasques. La primera part va ser alliberada el 1994 i la segona el 1996.

MPI són les sigles de **M**essage **P**assing **I**nterface.

MPI és ara l'estàndard pel pas de missatges i existeixen moltes APIs que l'implementen. Un exemple són Open MPI, Intel MPI, MPICH2, MVAPICH2.

Un tutorial complet es pot llegir a LLNL: *https://computing.llnl.gov/tutorials/mpi/*

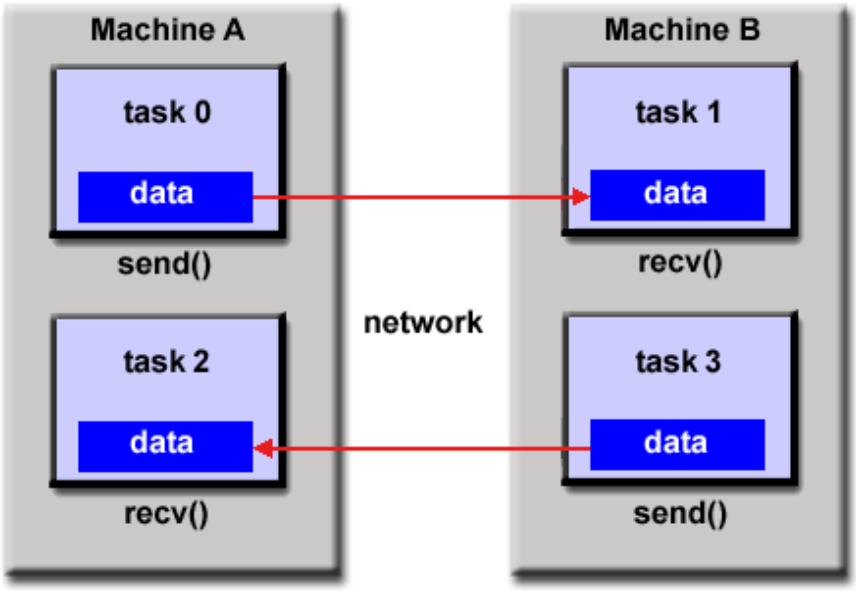


Figura 60: Model de programació de pas de missatges

Exemple de Hello World! executat amb 8 nodes:

```
/*
 * Sample MPI "hello world" application in C
 */

#include <stdio.h>
#include <mpi.h>

int main(int argc, char* argv[])
{
  int rank, size;

  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Comm_size(MPI_COMM_WORLD, &size);
  printf("Hello, world, I am %d of %d\n", rank, size);
  MPI_Finalize();

  return 0;
}

[user@acuario]$ mpirun -N 8 ./hello
Hello, world, I am 0 of 8
Hello, world, I am 1 of 8
Hello, world, I am 2 of 8
Hello, world, I am 4 of 8
Hello, world, I am 5 of 8
Hello, world, I am 6 of 8
Hello, world, I am 7 of 8
Hello, world, I am 3 of 8
```

4.3.3 Conclusió

Existeixen altres models de programació però gairebé tots es basen en els dos explicats. Hem de tenir en compte que en l'entorn CIMNE els codis normalment són de tipus sèrie o OpenMP, i que poca gent programa amb MPI.

En un primer cas disposem de codi sèrie sense paral·lelitzar. Seria el cas en que s'aprofitarien al màxim el nombre de processadors i el nombre de tasques de diferents usuaris concurrents en un mateix node. Des del punt de vista de l'administrador podríem tenir fins a 8 usuaris diferents treballant en els nodes M600 i M605 cadascun amb un procés, i fins a 12 en els nodes M610. No obstant podrien aparèixer problemes amb amples de banda de memòria.

En el cas de OpenMP estaríem aprofitant que els processadors són multi-nucli i que tenen memòria compartida. En el cas de disposar de dos processadors en una mateixa placa base, cas que ens ocupa en tots els nodes, es pot veure afectat negativament el rendiment per haver de comunicar-se amb la memòria de l'altra CPU. És important per tant saber en quantes parts dividim el codi i en quins processadors i nuclis es reparteixen, i ser més o menys conscients de l'ús que es fa de la memòria. Aquesta tasca correspon tant al programador com al kernel i també al planificador de treballs que implementem.

En el cas de MPI aprofitaríem que disposem de diversos nodes per realitzar una programació en memòria distribuïda. En aquest cas les comunicacions seran realitzades aprofitant l'ample de banda de l'Infiniband.

En un tercer cas podríem fer servir un model hibrid de programació que inclogués tasques MPI distribuïdes en diversos nodes i on cada tasca fos dividida mitjançant OpenMP. D'aquesta manera aconseguiríem el màxim grau de fragmentació del codi. Figura 61.

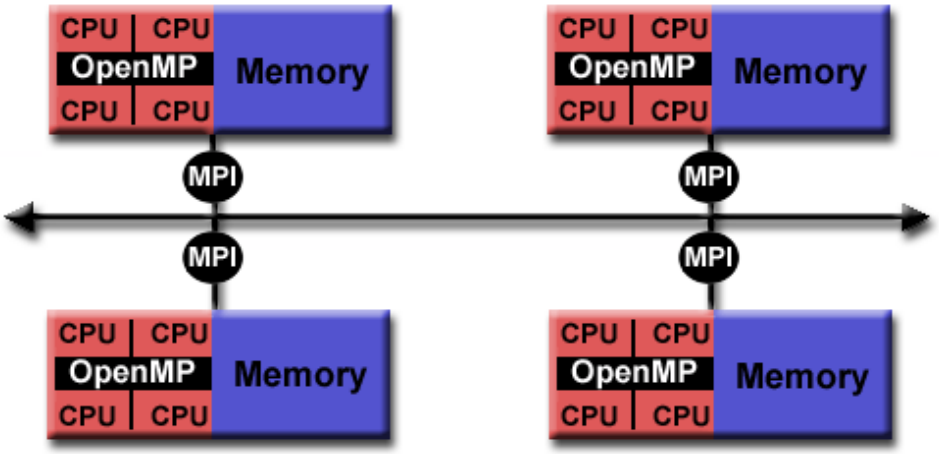
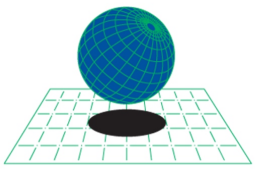


Figura 61: Model de programació hibrid, OpenMP + MPI



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 5
Implementació

5.1 Detall de la solució escollida

Com a resultat de la investigació realitzada en el Capítol 4 hem determinat quina serà la solució que millor encaixa a CIMNE i a les seves necessitats.

La solució escollida serà la de implementar una pila de software gratuïta i amb programari lliure en format manual (veure apartat 4.2.6), amb el sistema operatiu Scientific Linux 6.0 basat en RHEL 6.0 amb tots els components vistos a 4.2.3.1. Farà servir el sistema d'autenticació d'usuaris NIS. Com a gestor de recursos i planificador de tasques farem servir SLURM, i el servei de desplegament d'imatges serà gestionat per Kickstart de RedHat.

Descrivim i justifiquem a continuació cada decisió.

5.1.1 Format de la solució

Hem escollit una solució gratuïta i lliure per una banda per motius econòmics. Un dels objectius d'aquest projecte és el de reduir els costos possibles en la implementació i per tant no era una alternativa contractar a una empresa externa per tal que ens realitzessin el treball ni pagar per sistemes propietaris.

Per altre banda el motiu de voler realitzar la tasca manualment enlloc de en format paquet ha estat que sembla ser més complicat actualitzar i mantenir el sistema si la instal·lació no s'ha realitzat manualment. A més un paquet de software complet ens abstreu de tota la implementació i perdem el control i coneixement sobre el que s'està fent. Recordem que com a objectiu d'aquest projecte hi havia el conèixer en profunditat la implementació i els seus components.

Per aquestos motius s'escollirà el mètode manual amb programari lliure i gratuït.

5.1.2 Equipaments

Els equipaments a utilitzar seran el chassis, la cabina de discs SAN, els 16 nodes dels que disposem i el switch Dell.

Els servidors Vega i XFire hem determinat que han quedat obsolets i el departament recolza la decisió de reciclar aquests servidors o estudiar-ne un ús diferent al que tenen en aquest moment. D'aquesta manera podrem reduir l'energia del servei fins a 4800w menys. Per donar força a l'argument, l'estudi d'utilització del servei ha conclòs que són utilitzats per un nombre molt reduït d'usuaris i que aquests seran migrats al clúster.

5.1.3 Arquitectura del clúster

Capa 1 - Sistema Operatiu

El sistema operatiu escollit ha estat Scientific Linux 6.0.

Hem determinat que instal·laríem una solució basada en Linux ja que opcions alternatives com Windows, Solaris, etc. no són assequibles i a més poc utilitzades tal com veiem a la Figura 62,

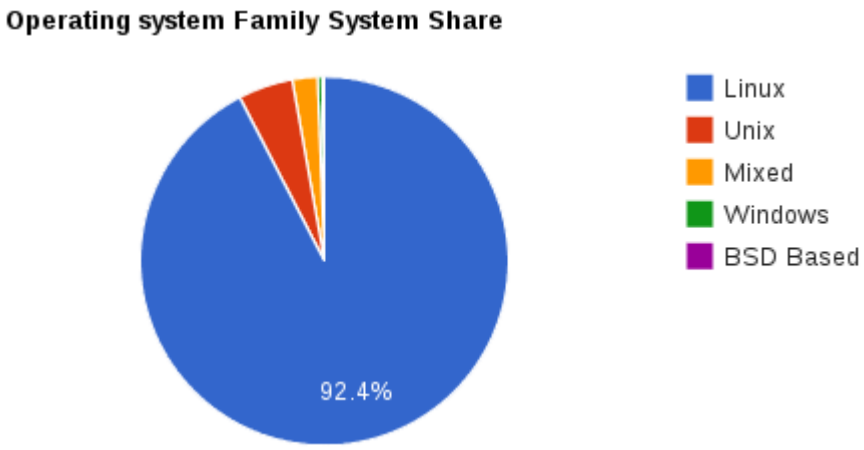


Figura 62: Estadística de sistema operatiu en el Top500 de Juny de 2012

Comparant les distribucions base que podrien ser més adients hem tingut en compte RedHat, Debian i Suse pel seu grau d'utilització. Partint de que les tres són distribucions de qualitat i que amb qualsevol d'aquestes es pot arribar a fer les mateixes funcions, s'ha determinat que:

De les tres distribucions hem descartat Suse per el motiu de comoditat d'ús. Freqüentment la comoditat de l'administrador amb una distribució o una altra és un factor que es considera com a important per escollir el sistema. En aquest cas la meva comoditat i experiència amb Suse no era tanta com amb el sistema Debian o RedHat.

Entre RedHat i Debian s'han comparat els usos més freqüents d'aquestes distribucions. En el món del HPC sembla a ser que RedHat té més influència que Debian segons dades del Top500 [44].

A més el software relacionat amb la capa de serveis de clúster habitualment té major suport per distribucions de tipus RedHat. Per altra banda la documentació d'aquesta en quant a HPC és més abundant que en el cas de Debian.

En quant a la distribució a utilitzar, els preus de RedHat Enterprise Linux consultats per una instal·lació de les nostres dimensions és massa elevada, arribant a 79$/node + 1598$ master / any a més de 199$ / node si disposem de xarxa d'altes prestacions. Si disposem de 15 nodes + 1 màster tenim que el preu anual és de 5768$ amb servei de suport estàndard. Es pot consultar el detall a la pàgina web de RedHat [45].

Al haver descartat RedHat com a sistema operatiu degut al seu elevat preu, hem valorat diferents alternatives cercant sempre compilacions a partir d'aquest:

1. CentOS: Tot i ser una distribució molt utilitzada per servidors i HPC, en el mes d'Abril de 2011 quan es va prendre aquesta decisió CentOS es trobava en un moment complicat. La versió de RedHat era la 6.0 des de feia uns mesos i CentOS havia quedat estancada a la 5.3, no havent-hi actualitzacions fins el moment. El seu futur no era clar i un dels objectius era instal·lar un sistema estable i suportat. Posteriorment al moment de la redacció d'aquest document CentOS s'ha recuperat i torna a ser una distribució puntera.
2. Fedora Core: És una distribució basada en RedHat EL però per entorn d'escriptori. No és per tant adient per instal·lar un clúster de càlcul per possibles problemes amb drivers, software, etc.
3. Scientific Linux: Avalada pel CERN (*http://public.web.cern.ch/public/*) i el Fermilab (*http://www.fnal.gov/*) a banda de diversos laboratoris i universitats del món, es centra en disposar d'una distribució adequada a la ciència. Un de les seus objectius és mantenir-la 100% compatible amb RHEL, i actualitzada en concordança el més ràpid possible:

   "Our main goal for the base distribution is to have everything compatible with Enterprise, with only a few minor additions or changes." [46]

   En el moment de l'elecció Scientific Linux es trobava a la versió 6.0, mateixa que la RHEL.

   Un altre avantatge és que tota la documentació de RedHat seria també vàlida per aquesta distribució.

Capa 2 – Seguretat i monitoreig

Utilitzarem els següents components per tal d'assegurar el control total del clúster.

- RAID 1 al node màster mitjançant la controladora hardware.
- Firewall iptables integrat a SL 6.0, amb regles de Input als ports SSH i Web, i Output només a la xarxa Infiniband i Ethernet interna.
- Accés SSH i limitació d'intents per usuari.
- Software DenyHosts per el control d'atacs per força bruta al port SSH.
- Política de contrasenyes PAM reforçada a 8 caràcters i 3 classes de caràcter.
- Sistema Bacula de backups integrat amb el robot de cintes del CPD, a més d'un sistema intern al SAN.
- Modificació de la quantitat de memòria per usuari usable al node màster, mitjançant /etc/security/limits.conf i altres possibles directives del sistema
- Instal·lació de Dell OpenManage (OMSA 6.0) per controlar les funcions de iDRAC, paràmetres de BIOS, actualitzacions de Firmware del hardware, etc.
- Instal·lació d'un navegador compatible per accedir a la interfície web del Dell Chassis Management Controller
- Instal·lació de Icinga client, succedani de Nagios per tal d'integrar-lo al sistema de monitoreig de CIMNE.

Capa 3 – Serveis de Clustering

Com a gestor de recursos i planificador de treballs hem escollit SLURM La decisió ha estat presa veient com en realitat tots els gestors de recursos analitzats proveeixen de característiques molt similars, essent la principal diferència els preus, la documentació disponible, el suport i les interfícies d'usuari i administrador.

Al estar situats al Campus Nord de la UPC i per proximitat al BSC (*http://www.bsc.es/*) molts investigadors de CIMNE disposen de compte als seus servidors. Al Marenostrum per exemple fan servir una solució de SLURM + Moab i per tant tenim referències properes del seu bon funcionament.

Per altre banda que el codi sigui lliure ens dona tranquil·litat si en el futur se'ns demana implementar alguna funcionalitat al servei o revisar algun problema.

Al realitzar una visualització a la documentació de Slurm hem vist que és molt entenedora i que està molt treballada, a més la comunitat ens ha assistit en alguns dubtes molt ràpidament.

Finalment decidim instal·lar només SLURM ja que aquest disposa d'un planificador intern senzill i actualment per les necessitats observades no en requerim un d'extern com Moab o Maui. Si fos necessari en el futur podria ser instal·lat ja que SLURM és compatible.

Els requisits de SLURM per la instal·lació requereixen els components següents:

- Sincronisme entre tots els usuaris del clúster: La primera alternativa fou la de fer servir el sistema LDAP de CIMNE i aprofitar la unificació de la gestió de que disposa el departament. No obstant problemes amb la configuració d'aquest que encara no estan resolts, com que tots els usuaris són d'un mateix grup, dificulten la implementació al clúster. Per aquest motiu s'ha decidit inicialment d'instal·lar el servei NIS i utilitzar usuaris locals, tenint sempre la possibilitat en el futur de canviar a un sistema diferent.
- Sincronisme de temps: S'instal·larà un servidor NTP al node màster i un client d'aquest a tots els nodes. D'aquesta manera el temps del clúster serà el mateix a tot el sistema.
- Servei de desplegament d'imatges a nodes: Hem optat per la opció de fer servir la opció més estàndard de RedHat, que és Kickstart. Probablement qualsevol solució hagués funcionat d'igual manera però aquesta és senzilla de configurar i eficaç. Com a suport necessitem un servidor de DHCP, TFTP i PXE.
- Repositori de software global: Per tal d'unificar tot el software del clúster i acomplir amb l'objectiu d'aconseguir un sistema uniforme, configurarem un repositori de paquets RPM per utilitzar amb YUM on apuntaran totes les fonts dels nodes i del node màster. Aquest repositori serà sincronitzat periòdicament amb els oficials. També contindrà un directori de paquets privat on hi col·locarem paquets personalitzats per nosaltres.
- Espai compartit NFS: Per tal de proporcionar un directori principal per cada usuari on pugui realitzar els seus càlculs i guardar temporalment els resultats, es crearà un espai compartit NFS que serà compartit amb tots els nodes. L'espai serà obtingut per un volum iSCSI de la cabina de discs SAN. S'aplicaran quotes de disc als usuaris.

  També es proporcionarà un directori compartit a /opt per el software que ha d'estar disponible a tots els nodes i que no és instal·lable mitjançant YUM.

  Finalment un espai compartit per configuracions idèntiques a tots els nodes serà muntat al servidor.

  Es pot veure en detall l'esquema de directoris a l'apartat 5.1.6 Organització del sistema de fitxers.

Capa 4 – Interfícies d'administrador & usuari

Com a eines d'administrador hem instal·lat el següent:

- Benchmarks & tests: Hem desenvolupat petits programes amb OpenMP i MPI que mostren la comunicació entre nodes. També s'executaran el “Regression Test”, una suite de tests que venen amb el propi SLURM i que ens assegurarà que tots els components són funcionals i estan instal·lats correctament.
- Gestió de paquets: El software s'instal·larà a ser possible amb YUM al node màster i a tots els nodes si fos necessari. En casos d'haver d'instal·lar software que no es troba en format RPM es situarà a /opt.
- Eines de gestió global: Instal·larem el paquet C3 i principalment s'utilitzarà l'eina “cexec” que permetrà executar comandes a tots els nodes al mateix temps.

Com a eines d'usuaris instal·larem:

- Servidor SSH al node màster per accés remot.
- Editors: Emacs, Vim, Nano, Gedit
- Intèrprets: Bash, TCSH
- Software: GiD, Matlab, Make, Cmake, Intel Compiler, GCC, Valgrind, Python, GNUPlot.
- Biblioteques: OpenMPI, Intel MPI, Mpich2, Mvapich, Mvapich2, Boost, OpenMP, Blas, Lapack, Swig, Papi, Intel MKL.
- Configuració de la eina “environment-modules” per tal de facilitar la càrrega de variables d'entorn a l'usuari.

5.1.4 Gestor de recursos & Planificador

5.1.4.1 Planificador de treballs

S'instal·larà SLURM amb un planificador de tasques de tipus backfill. Per fer aquesta elecció s'ha consultat als investigadors i les conclusions respecte als altres planificadors han estat les següents:

- FIFO: Aquesta és la opció més senzilla i els investigadors estaven d'acord amb ella. El fet que finalment no s'hagi elegit ha estat perquè el Backfill proporciona també la funcionalitat FIFO però amb la característica afegida de poder inserir treballs en forats buits.
- Gang schedule: Aquesta política permet parar i iniciar treballs mentre estan funcionant i maximitza l'ús del cluster. Des del punt de vista de l'administrador és una bona política ja que redueix molt el temps de inactivitat dels recursos, i per part de l'empresa sembla que ha de motivar un major rendiment al estar els equips parats menys temps. No obstant els investigadors no hi han estat d'acord ja que sovint realitzen proves d'escalat dels seus codis i aquesta política permet que siguin interromputs de formes inesperades, evitant així l'escalat correcte del codi.
- Preemption: Aquesta política ha agradat a certs investigadors però altres s'han oposat totalment. Recordem que es beneficia a treballs amb més prioritat, i si fos el cas que un usuari fes servir molt el clúster i li baixés la prioritat es veuria molt perjudicat. Realment és una política equitativa però també sofreix el problema del Gang.
- Backfill: Aquesta sembla ser la més adequada per el nostre cas i serà la que s'escollirà. El motiu és que és una FIFO però amb l'afegit de poder inserir treballs en forats buits. Aquesta característica ens maximitzarà en el possible l'ús del clúster mentre s'especifici un temps als treballs. Per tant haurem d'obligar a posar uns temps màxims als treballs predefinits per obligar a l'usuari a modificar aquests temps.

5.1.4.2 Prioritats dels treballs

S'ha determinat fer servir el plugin “multi-factor” de SLURM que permet determinar la prioritat del treball amb molts factors que ja hem comentat anteriorment. L'afinat d'aquest plugin s'haurà d'anar fent en funció dels resultats. Per activar-lo necessitarem activar també l'accounting i per tant instal·larem una base de dades MySQL que emmagatzemarà tota la informació dels usuaris i els seus treballs.

5.1.4.3 Particions de nodes (cues)

Després de realitzar proves, comprovar com es realitzava l'activitat, i recollir opinions les particions que es realitzaran separaran els nodes de cada arquitectura (M600, M605 i M610) i a més deixaran dos nodes M600 amb un temps màxim de càlcul de 1 dia. Les altres tindran un temps màxim de càlcul de 90 dies.

```
PartitionName:    Main        Nodes=pez[001-008]     Default=YES MaxTime=90-0
PartitionName:    Short       Nodes=pez009,pez010     Default=NO  MaxTime=1-0
PartitionName:    AMD2356     Nodes=pez011,pez012     Default=NO  MaxTime=90-0
PartitionName:    XeonE5645   Nodes=pez0[13-15]       Default=NO  MaxTime=90-0
```

Finalment per problemes de bus de memòria en els nodes M600, on els programes dels investigadors no escalen correctament amb més de 3 fills, s'ha decidit limitar-los a 2 processos per socket.

5.1.5 Estructura de xarxa

Noms de host

El clúster que muntarem disposa de 1 node màster anomenat Acuario i 15 nodes esclau que anomenarem peixos. El nom dels peixos seran pez[001-015].

Acuario disposarà d'una IP pública amb accés a Internet amb el nom acuario.cimne.upc.edu.

Adreçament

La separació de les xarxes es farà mitjançant diferents VLAN, corresponent a cadascuna un rang d'adreces diferenciat:

eth iSCSI (10.50.0.0/24) - Xarxa destinada a l'emmagatzemament iSCSI. A aquesta xarxa s'hi connectaran els dispositius d'emmagatzemament com la cabina SAN MD3000i.

eth Nodes (10.0.1.0\24) - Xarxa de connexió ethernet destinada a la comunicació entre nodes i Acuario. S'utilitzarà per les comunicacions que requereixin poc rendiment.

Infiniband Nodes (10.0.0.0\24) - Xarxa Infiniband. S'utilitza per el pas de missatges durant els càlculs dels científics i exportació de sistemes de fitxers usats per càlcul.

eth Management (172.26.0.0\24) - Xarxa ethernet destinada a l'iDrac i a les interfícies i dispositius de gestió, per exemple el Dell CMC o el port de gestió del SAN.

Internet (147.83.143.0\24) - Xarxa ethernet amb connexió a Internet, només per el node màster.

L'adreçament d'aquestes VLAN es realitzarà per cada node afegint el seu sufix a la xarxa, per exemple per el pez006 la IP pertanyent a la eth Nodes serà: 10.0.1.6, i la pertanyent a la Infiniband la 10.0.0.6.

L'excepció serà Acuario que disposarà de les adreces 10.50.0.100, 10.0.1.100, 10.0.0.100, 172.26.0.100 i 147.83.143.111 cadascuna a la seva respectiva vlan.

Als annexos següents disposem d'un esquema lògic de la xarxa amb les seves VLAN, una taula d'adreçaments IP i un diagrama de cablejat físic.

A.3.3 Esquema de la xarxa – VLANs

A.3.4 Adreçament IP

A.3.5 Cablejat de xarxa

5.1.6 Organització del sistema de fitxers

Definirem la següent estructura de directoris:

- / — Directori arrel on hi haurà els fitxers típics d'un sistema operatiu. Espai de 101GiB, disc local.
- /mnt/cluster-data — Directori muntat a un volum de la cabina de discs que contindrà configuracions, còpies de seguretat, el repositori de paquets, etc. Espai aproximat: 70GiB.
- /data0 — Punt de muntatge idèntic a /mnt/cluster-data per adequar el sistema a l'estructura de backups de CIMNE.
- /home — Directori muntat a un volum de la cabina de discs que contindrà els directoris principals dels usuaris. Espai aproximat: 10TiB. Quotes per usuari a especificar més endavant.

L'estructura més important es troba dins /mnt/cluster-data i és la següent:

/mnt/cluster-data	
|-- backups	*Directori de backups*
|-- logs	· Aquests dos directoris contenen l'esquema que segueix
`-- rsync	CIMNE per realitzar els seus backups.
|- globalfs	*Directori compartit a tots els nodes*
|- bin	· Fitxers binaris o scripts compartits a tot el clúster
|- etc	· Fitxers de configuració compartits al llarg del clúster
|- first-boot	· Fitxers script per configurar el primer arranc dels nodes
`- opt	· Software compartit al llarg dels node
|- kickstart	Directori per els fitxers del desplegament d'imatges
|- repo	*Repositori de fitxers YUM*
|- sbin	*Scripts personalitzats per l'administrador*
`- tftpboot	*Protocol PXE+TFTPBOOT*

El punt de muntatge / contindrà els següents directoris i enllaços rellevants:

/	
|-- globalfs	*Enllaç simbòlic a /mnt/cluster-data/globalfs*
|-- kickstart	*Enllaç simbòlic a /mnt/cluster-data/kickstart*
|-- repo	*Enllaç simbòlic a /mnt/cluster-data/repo*
|-- tftpboot	*Enllaç simbòlic a /mnt/cluster-data/tftpboot*
|-- opt	*Directori amb el software addicional instal lat al node màster*
|-- /home	*Directori personal dels usuaris*
`- root	*Directori de l'usuari root*

5.1.6.1 Còpies de seguretat amb el sistema de fitxers previst

Gràcies a la organització de fitxers que hem determinat tota la informació que caldria per recuperar-nos d'un desastre es podria trobar dins:

- /mnt/cluster-data

a excepció de /etc, /var/lib, /var/www i /opt del node màster. Llavors crearem un cron i un script per tal que es faixin còpies d'aquests directoris periòdicament dins /mnt/cluster-data/backups/rsync. Determinarem la periodicitat d'aquests perquè es realitzin cada 2 dies començant per dilluns.

L'script anirà dins els binaris d'administrador, a /mnt/cluster-data/sbin/backup-system.sh.

Es definiran també còpies periòdiques a Gollum, el servidor de backups de CIMNE. Aquest servidor agafarà i traspassarà a cintes el directori sencer /data0.

5.2 Instal·lació

5.2.1 Cablejat de xarxa i configuració del Switch

Per començar la instal·lació del sistema és necessari connectar de forma correcta tots els components. Hem desenvolupat un gràfic de connexió explicatiu on es detalla quines connexions realitzar i a quins ports. S'ha de seguir aquest gràfic i implementar-ho al clúster abans de continuar. Una única puntualització a l'esquema determinat és que si en el futur no es preveu que la cabina de discs vagi connectada a cap altre servidor que no sigui el node màster Acuario, es pot alliberar el pas per el switch i connectar directament els dos ports al node. A efectes pràctics no té cap importància més que s'allibera al switch de tràfic. Veure A.3.5 Cablejat de xarxa.

Un cop cablejat el sistema és moment de configurar el switch PowerConnect 5424. Connectarem un cable sèrie amb un portàtil i mitjançant telnet entrarem a la interfície inicial de configuració. Executarem la configuració inicial amb l'assistent i que es descriu al manual d'usuari [doc18]. Donarem la IP 172.26.0.202 tal com definirem al punt 5.1.5 Estructura de xarxa.

L'usuari que donarem serà *admin*, i la contrasenya la de Sistemes CIMNE. Configurarem també l'accés per HTTP i per SSH seguint el manual. Per exemple per l'accés HTTP:

```
console> enable
console# configure
console(config)# ip http authentication local
console(config)# username admin password **** level 15
```

Una vegada configurat, podrem accedir amb una interfície web més intuïtiva. Des d'aquesta interfície configurarem les diferents VLAN que hem especificat a 5.1.5 Estructura de xarxa i activarem la optimització iSCSI.

A la Figura 63 podem veure la interfície web de configuració a la pantalla de optimització de iSCSI. El switch ha detectat automàticament les sessions iSCSI entre Acuario i el SAN.

A la Figura 64 mostrem la pantalla de configuració de les VLAN.

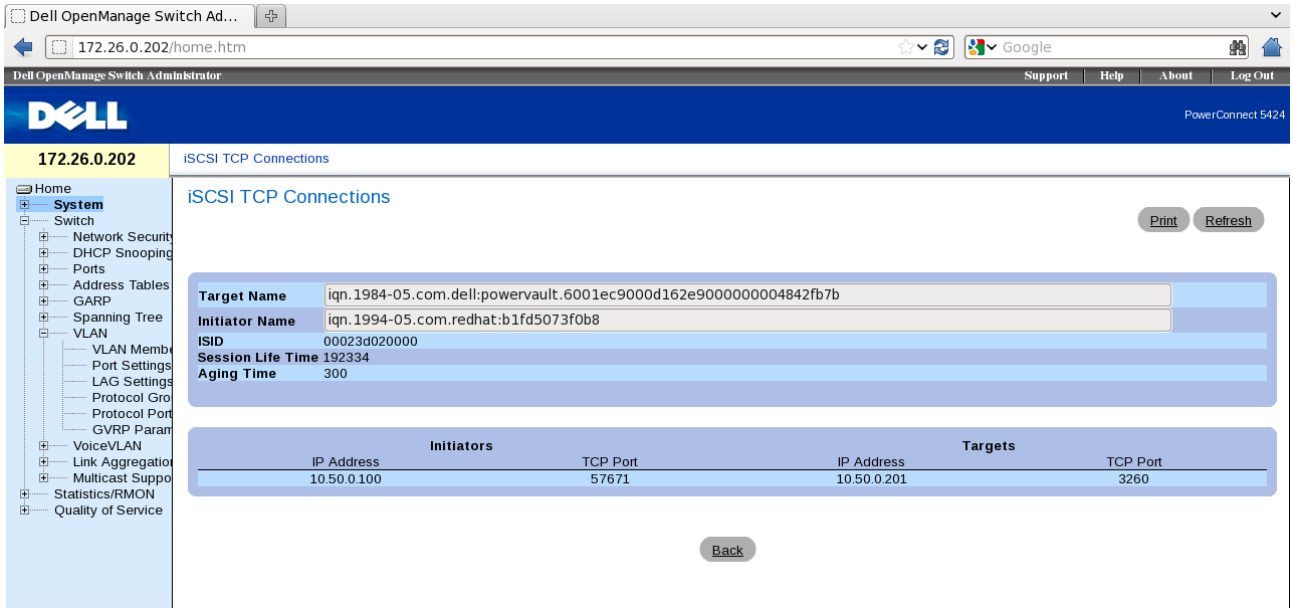


Figura 63: El switch PowerConnect reconeix sessions iSCSI

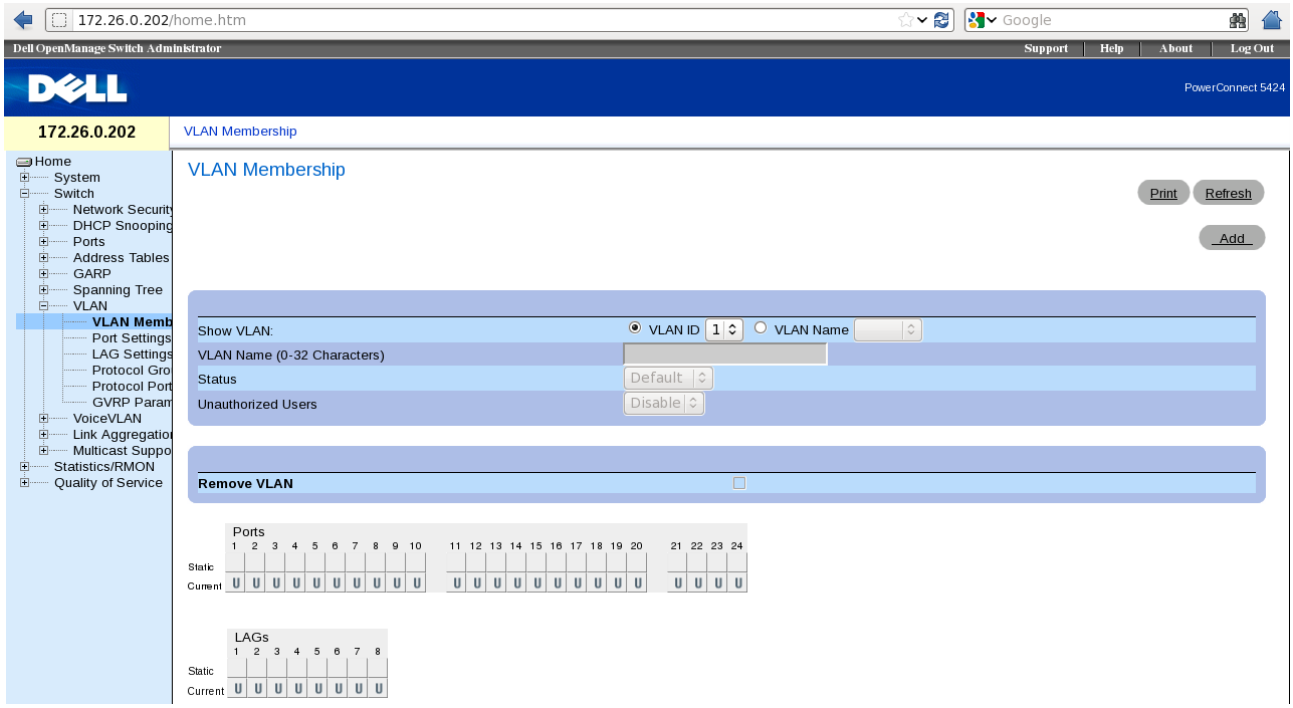


Figura 64: Configuració de les VLAN al PowerConnect 5424

5.2.2 Capa 1 – Sistema operatiu

1. Realitzem un RAID 1 per hardware al node màster.
2. Iniciem amb el DVD de Scientific Linux 6.0 que haurem baixat de la seva pàgina web.
3. Instal·lem el sistema operatiu amb les següents opcions:

   Configuracions:

Hostname:	Acuario
Idioma:	anglès, en_US.UTF-8
Zona:	Europe/Madrid
Teclat:	espanyol
Xarxa:	dispositiu eth1, per dhcp
Instal·lació:	CD-ROM
SELinux:	Desactivat
Firewall:	Desactivat

   És important desactivar SELinux.

4. La sel·lecció de paquets serà la següent, obtinguda del log d'instal·lació d'anaconda (/root/anaconda-ks.cfg):

```
%packages
@base
@catalan-support
@console-internet
@core
@debugging
@basic-desktop
@emacs
@graphical-admin-
tools
@hardware-monitoring
@infiniband
@internet-browser
@java-platform
@large-systems
@nfs-file-server
@network-file-system-
client
@performance
@perl-runtime
@scientific
@server-platform
@spanish-support
@storage-client-
multipath
@system-admin-tools
@storage-client-iscsi
mtools
pax
squashfs-tools
sgpio
genisoimage
tigervnc-server
emacs-nox
system-config-lvm
system-config-
kickstart
lm_sensors
qperf
perftest
infiniband-diags
srptools
opensm
sg3_utils
perl-DBD-SQLite
mpitests-openmpi
mvapich2
mpich2
mpitests-mvapich
mpitests-mvapich2
openmpi
atlas
mvapich
numpy
lsscsi
-libipathverbs
-totem-mozplugin
-nspluginwrapper
-units
-gnuplot
```

Tots els drivers necessaris del sistema ja han estat seleccionats per aquesta instal·lació.

5.2.2.1 Configuració de la xarxa

Configurarem la xarxa seguint l'esquema que hem comentat a 5.1.5.

1. Primer de tot hem de seleccionar l'eina que vulguem utilitzar per realitzar les configuracions. Per defecte ve instal·lat el servei NetworkManager que tot i ser una bona eina per usuaris novells, a nosaltres ens anirà millor el servei “network”.

Desactivem per tant NetworkManager i activem el servei network:

```
chkconfig NetworkManager off
chkconfig network on
service NetworkManager stop
service network start
```

2. A continuació ens assegurem que el firewall no està activat ja que per realitzar les proves inicials ho farem sense. El configurarem en etapes posteriors.

```
chkconfig iptables off
chkconfig ip6tables off
service iptables stop
service ip6tables stop
```

3. A continuació ens fa falta instal·lar el paquet “rdma” que s'encarrega de gestionar el llançament de la xarxa Infiniband. Igualment, com que estem instal·lant el node Master del Cluster, aquest s'encarregarà de fer d'encaminador de comunicacions Infiniband. Necessitarem per tant el servei opensm.

```
yum install rdma
chkconfig rdma on
service rdma start
chkconfig opensm on
service opensm start
```

4. Per configurar les interfícies de xarxa hem de modificar els fitxers que es troben dins el directori /etc/sysconfig/network-scripts.

S'ha de tenir present que /etc/sysconfig/networking/devices/, /etc/sysconfig/networking/profiles/ i /etc/sysconfig/network-scripts/ contenen fitxers ifcfg-* enllaçats amb hard-links.

L'eina network només utilitza els que es troben dins /etc/sysconfig/network-scripts/ i només ens hem de preocupar dels altres directoris si es fa servir l'eina NetworkManager.

Posem a continuació els fitxers de configuració que s'han de crear per cada interfície:

```
[root@acuario network-scripts]# cat ifcfg-eth0
DEVICE=eth0
NM_CONTROLLED=yes
ONBOOT=yes
IPADDR=10.0.1.100
BOOTPROTO=none
NETMASK=255.255.255.0
TYPE=Ethernet
IPV6INIT=no
USERCTL=no
HWADDR=00:1E:C9:CD:2D:EF
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
NAME="System eth0"
```

```
[root@acuario network-scripts]# cat ifcfg-eth0:0
DEVICE=eth0:0
NM_CONTROLLED=yes
BOOTPROTO=none
IPADDR=172.26.0.100
NETMASK=255.255.255.0
ONBOOT=yes
IPV6INIT=no
HWADDR=00:1E:C9:CD:2D:EF
USERCTL=no
NAME="System eth0 Alias"

[root@acuario network-scripts]# cat ifcfg-eth1
DEVICE=eth1
NM_CONTROLLED=yes
ONBOOT=yes
HWADDR=00:1e:c9:cd:2d:f1
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth1"
IPADDR=147.83.143.111
NETMASK=255.255.255.0
DNS2=8.8.8.8
DNS1=147.83.143.133
USERCTL=no
GATEWAY=147.83.143.1

[root@acuario network-scripts]# cat ifcfg-eth2
DEVICE=eth2
NM_CONTROLLED=yes
USERCTL=no
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:1D:09:72:25:84
MASTER=bond0
SLAVE=yes
IPV4_FAILURE_FATAL=yes
NAME="System eth2"

[root@acuario network-scripts]# cat ifcfg-eth3
DEVICE=eth3
NM_CONTROLLED=yes
USERCTL=no
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:1D:09:72:25:86
MASTER=bond0
SLAVE=yes
IPV4_FAILURE_FATAL=yes
NAME="System eth3"
```

```
[root@acuario network-scripts]# cat ifcfg-bond0
DEVICE=bond0
NM_CONTROLLED=yes
USERCTL=no
ONBOOT=yes
BOOTPROTO=none
IPADDR=10.50.0.100
NETMASK=255.255.255.0
TYPE=Ethernet
IPV6INIT=no
IPV4_FAILURE_FATAL=yes
NAME="Redundant eth3 i eth4"
BONDING_OPTS="mode=balance-alb miimon=100"

[root@acuario network-scripts]# cat ifcfg-ib0
DEVICE=ib0
NM_CONTROLLED=yes
ONBOOT=yes
IPADDR=10.0.0.100
BOOTPROTO=none
NETMASK=255.255.255.0
NAME="System ib0"
TYPE=Ethernet
```

En el cas de les interfícies eth2 i eth3, podem veure com tenen una configuració especial declarant-se esclaves de la interfície bond0. Aquesta última al mateix temps disposa d'unes opcions de bonding (*BONDING_OPTS="mode=balance-alb miimon=100")* que especifiquen els algorismes de balanceig i altres opcions interessants.

A més per fer que la interfície de bonding carregui correctament, ens caldrà crear el fitxer /etc/modprobe.d/bonding.conf amb el contingut següent:

```
alias netdev-bond0 bonding
```

5. Configurarem finalment el DNS i el nom de host d'Acuario:

```
[root@acuario network-scripts]# cat /etc/hosts
127.0.0.1       localhost.localdomain localhost
::1             localhost6.localdomain6    localhost6
147.83.143.111  acuario.cimne.upc.edu
10.0.0.100      ib_acuario
10.0.1.100      eth_acuario acuario
```

El fitxer de DNS pot apuntar a qualsevol servidor de noms que tinguem com a preferit, per exemple:

```
[root@acuari network-scripts]# cat /etc/resolv.conf
nameserver 147.83.143.133
nameserver 8.8.8.8
```

6. Reiniciem el servidor i comprovem que totes les IP's s'hagin assignat correctament i hi hagi connectivitat amb altres punts de la xarxa.

Per comprovar que un dels dos HCA Inifniband està connectat (la segona és la que proporciona un port extern, que per ara no utilitzem), executarem el següent comprovant que un dels dos ports està amb estat “ACTIVE”:

```
[root@acuari ~]# ibstat
CA 'mlx4_0'
      CA type: MT25418
      Number of ports: 2
      Firmware version: 2.3.0
      Hardware version: a0
      Node GUID: 0x00188b9097fe14b5
      System image GUID: 0x00188b9097fe14b8
      Port 1:
            State: Active
            Physical state: LinkUp
            Rate: 20
            Base lid: 9
            LMC: 0
            SM lid: 9
            Capability mask: 0x0251086a
            Port GUID: 0x00188b9097fe14b6
      Port 2:
            State: Down
            ....
```

Si recordem la codificació de l'infiniband DDR 4x definia un ratio de 20Gbps total. El rendiment real era del 80% amb la codificació 8B/10B de payload/dades totals. Aquesta rendiment limita la velocitat pic real a 16Gbps [doc15].

5.2.2.2 Configuració iSCSI

La configuració iSCSI serveix per connectar-nos a la cabina de discs SAN. Prèviament haurem d'haver creat els volums per /home i /mnt/cluster-data. S'explicarà a l'apartat d'instal·lació del MD Storage Manager 5.2.3.8.

Per configurar l'iSCSI hem d'instal·lar el servei, activar-lo, i descobrir els targets. Al procés de descobrir els targets se li dona la IP de la cabina havent creat prèviament un volum al qual s'hagi donat accés a Acuario. És possible donar accés a un volum mitjançant IPs o contrasenyes CHAP. En el cas d'haver donat accés per IP no cal fer res, d'altra manera, per accés CHAP cal configurar les contrasenyes d'accés dins /etc/iscsi/iscsid.conf.

```
yum install iscsi-initiator-utils
chkconfig iscsid on
chkconfig iscsi on
service iscsid start
service iscsi start
iscsiadm --mode discovery --type sendtargets --portal 10.50.0.200:3260
```

En cas de sorgir algun problema amb el modul de transport de iscsi, executar:

```
modprobe scsi_transport_iscsi
```

Finalment, una vegada descoberts els targets, se'ns haurà muntat un nou disc que podrem trobar amb:

```
tail -f /var/log/messages
sd 5:0:0:0: [sdc] 512-byte logical blocks: (9.99 TB/9.09 TiB)
....
sd 5:0:0:0: [sdc] Attached SCSI disk
```

En cas de que no ens surti cap disc nou, connectarem manualment al target amb el seu identificador iSCSI obtingut des del gestor del SAN o des de els targets descoberts:

```
iscsiadm -m node -T ip-10.50.0.200:3260-iscsi-iqn.1984-\
05.com.dell:powervault.6001ec9000d162e9000000004842fb7b-lun-1-part1\
10.50.0.200 --login
```

La línia identificadora com veiem conté la IP a la que ens connectem (10.50.0.200), el port (3260), el distribuïdor (dell) i el nom de la cabina (powervault...). Al final trobem el nombre de lun i la partició o nom de volum:

ip-**10.50.0.200**:3260-iscsi-iqn.1984-05.com.**dell**:**powervault**.6001ec90...04842fb7b-**lun-1-part1**

Una vegada vist el disc pel kernel amb dmesg bastarà formatar-lo i tractar-lo com un disc normal. Al reiniciar, automàticament es connectarà de nou el disc iSCSI.

En cas que no connecti automàticament, executar:

```
iscsiadm -m node -T <nom_del_volum> <ip_SAN> --op update -n
node.startup -v automatic
```

Si volem desconnectar (fer logout) d'un volum, primer desmuntar-lo i llavors executar:

```
iscsiadm -m node -T <nom_del_volum> <ip_SAN> –logout
```

A més, si el volum ha de tenir més de 2TB, haurem de fer servir parted per formatar-lo.

Per posar les noves entrades a l'fstab, es recomana no fer servir els noms /dev/sdX, ja que en posteriors inicis del sistema poden canviar. Enlloc d'això es convenient determinar quins volums volem muntar utilitzant la nomenclatura de /dev/disk/by-id/, on apareixeran els noms dels lun iSCSI.

Per exemple:

```
[root@acuario ~]# cat /etc/fstab
...
...
#SAN
/dev/disk/by-path/<nom_volum> /home        ext4 _netdev,defaults0 0
/dev/disk/by-path/<nom_volum> /mnt/cluster-data ext4 _netdev,defau..
/dev/disk/by-path/<nom_volum> /data0       ext4 _netdev,defaults0 0
...
```

Repetirem el mateix procediment tant pel volum de /home com pel de /mnt/cluster-data. Finalment a /etc/fstab duplicarem la línia de /mnt/cluster-data però muntant el sistema de fitxers a /data0. Ho fem per adaptar-nos a l'esquema de backups de CIMNE.

En cas de tenir problemes podem veure a quines sessions estem connectats amb la comanda:

```
[root@acuario iscsi]# iscsiadm -m session
tcp:[1]10.50.0.200:3260,1 iqn.1984-05.com.dell:powervault.6001ec9000d....0004842fb7b
tcp:[2]10.50.0.201:3260,1 iqn.1984-05.com.dell:powervault.6001ec9000d....0004842fb7b
```

Si no conseguim fer login obtenint algun error podem comprovar quines connexions hi ha definides amb el següent procediment:

```
[root@acuario ~]# cd /var/lib/iscsi/
```

```
[root@acuario iscsi]# find .
.
./slp
./nodes
./nodes/iqn.1984-05.com.dell:powervault.6001ec9000d162e9000000004842fb7b
./nodes/iqn.1984-
05.com.dell:powervault.6001ec9000d162e9000000004842fb7b/10.50.0.200,3260,1
./nodes/iqn.1984-
05.com.dell:powervault.6001ec9000d162e9000000004842fb7b/10.50.0.200,3260,1/default
./nodes/iqn.1984-
05.com.dell:powervault.6001ec9000d162e9000000004842fb7b/10.50.0.201,3260,1
./nodes/iqn.1984-
05.com.dell:powervault.6001ec9000d162e9000000004842fb7b/10.50.0.201,3260,1/default
./static
./ifaces
./send_targets
./send_targets/10.50.0.201,3260
./send_targets/10.50.0.201,3260/st_config
./send_targets/10.50.0.200,3260
./send_targets/10.50.0.200,3260/iqn.1984-
05.com.dell:powervault.6001ec9000d162e9000000004842fb7b,10.50.0.200,3260,1,default
./send_targets/10.50.0.200,3260/iqn.1984-
05.com.dell:powervault.6001ec9000d162e9000000004842fb7b,10.50.0.201,3260,1,default
./send_targets/10.50.0.200,3260/st_config
./isns
```

Podrem eliminar aquestes connexions amb la comanda:

```
iscsiadm -m discoverydb -t st -p <nom_de_connexio> -o delete
```

5.2.2.3 Quotes de disc

Un cop muntada la partició /home és moment de definir les quotes d'usuari. Ho farem de la forma habitual en sistemes Linux, afegint els paràmetres “usrquota” i “grpquota” a /etc/fstab pels punts de muntatge que ens interessin.

Llavors crearem la base de dades de quotes a l'arrel del punt de muntatge. Els fitxers s'anomenen “*aquota.user”* i *“aquota.group”.* Posteriorment reiniciarem o re-muntarem el sistema de fitxers.

```
[root@acuario ~]# quotacheck -cug /home
```

Posteriorment inicialitzarem la BBDD i comprovarem que la quota s'ha aplicat:

```
[root@acuario ~]# quotacheck -vgua
[root@acuario ~]# quotaon -av
```

I finalment podrem editar les quotes per cada usuari:

```
[root@acuario ~]# edquota -u nom_usuari
[root@acuario ~]# edquota -g nom_grup
```

Per llistar les quotes actuals:

```
[root@acuario ~]# quota -u nom_usuari
```

Podem trobar més detall de com realitzar el procés als documents de RHEL 6, Capítol 18. Disk Quotas, [50].

5.2.3 Capa 2 – Seguretat i monitoreig

5.2.3.1 Firewall

Per obtenir en detall la forma de configuració del firewall iptables es poden consultar els manuals de RHEL 6 [50]. Tenim dues opcions de configuració, modificant el fitxer /etc/sysconfig/iptables, o amb l'eina gràfica system-config-firewall, Figura 66.

Als nodes no hi instal·larem firewall.

Les regles que hem de crear al node màster les reflexem a la Taula 10.

Política	*Ports*	*Direcció*	*VLAN*	*Interfície*
ACCEPT	Tots	IN / OUT	Infiniband Nodes	ib0
ACCEPT	Tots	IN / OUT	eth Nodes	eth0
ACCEPT	Tots	IN / OUT	eth Management	eth0:0
ACCEPT	Tots	IN	eth iSCSI	bond0
ACCEPT	80, 9102, 22	IN	Internet	eth1
ACCEPT	9103	OUT	Internet	eth1
DENY	Tots	OUT - {SYN}	Internet	eth1
DENY	Tots	IN	Internet	eth1

Taula 10: Regles a implementar al firewall d' Acuario

Recordem el funcionament de les regles iptables a la Figura 65.

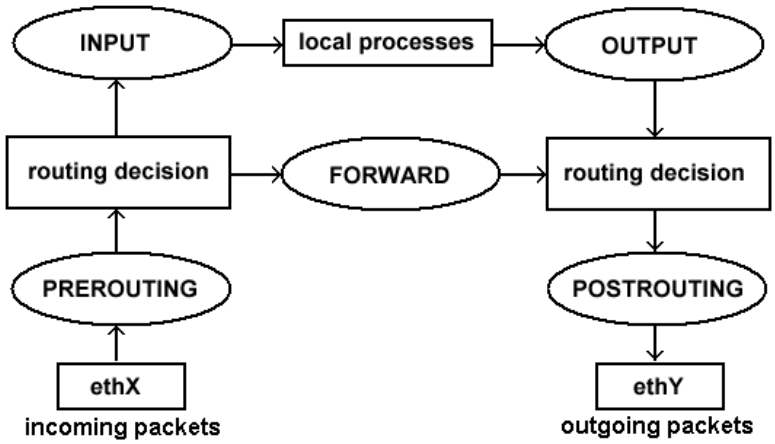


Figura 65: Funcionament de les regles iptables [51]

Executarem les següents ordres per configurar el firewall [52]:

```
iptables -I INPUT 1 -i lo -p all -j ACCEPT
iptables -I INPUT 1 -i ib0 -p all -j ACCEPT
iptables -I INPUT 1 -i eth0 -p all -j ACCEPT
iptables -I INPUT 1 -i eth0:0 -p all -j ACCEPT
iptables -I INPUT 1 -i bond0 -p all -j ACCEPT

iptables -I OUTPUT 1 -i lo -p all -j ACCEPT
iptables -I OUTPUT 1 -i ib0 -p all -j ACCEPT
iptables -I OUTPUT 1 -i eth0 -p all -j ACCEPT
iptables -I OUTPUT 1 -i eth0:0 -p all -j ACCEPT
iptables -I OUTPUT 1 -i bond0 -p all -j ACCEPT

iptables -A INPUT -i eth1 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

iptables -A INPUT -i eth1 -p tcp --dport 9102 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -p tcp --sport 9102 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -p tcp --dport 9103 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth1 -p tcp --sport 9103 -m state --state ESTABLISHED -j ACCEPT

iptables -A INPUT -i eth1 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT
iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Guardem llavors les regles al fitxer /etc/sysconfig/iptables:

```
service iptables save
```

Configurem l'inici del servei:

```
chkconfig --level 345 iptables on
service iptables start
```

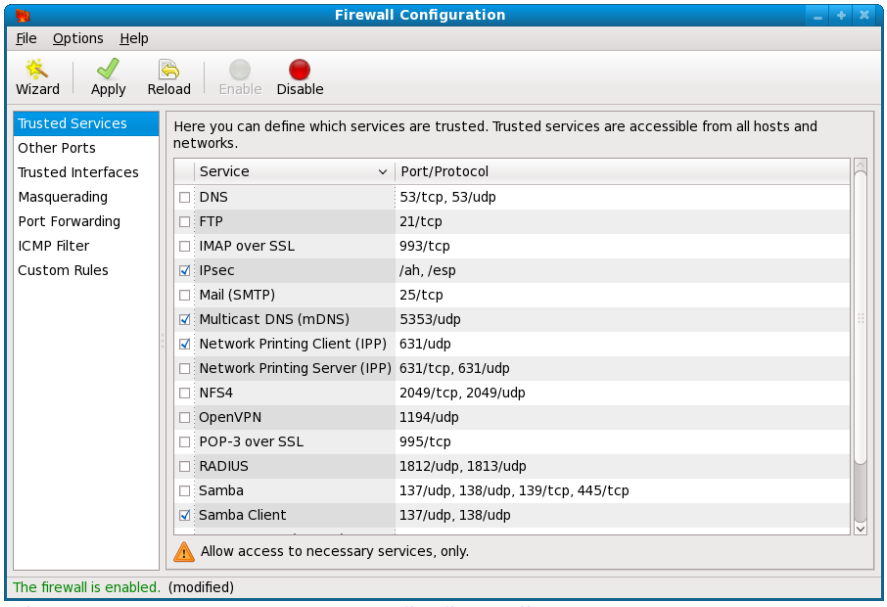


Figura 66: RHEL6 system-config-firewall

5.2.3.2 Servei SSH & Denyhosts

Per tal de permetre l'accés al sistema s'instal·larà el servei SSH i el paquet de software Denyhosts. Aquest últim és el que utilitza CIMNE per controlar els intents d'accés per força bruta. L'altre alternativa era la de donar accés per claus privades però això suposava una càrrega de gestió més elevada a més d'una complicació per els usuaris.

També haguéssim pogut bloquejar els intents per el propi firewall iptables però Denyhosts ens dona un control més acurat.

El procediment d'instal·lació i activació de SSH i Denyhosts és l'habitual per YUM, només recordar a activar els serveis a l'inici del sistema:

```
chkconfig sshd on
chkconfig denyhosts on
```

Respecte a la configuració de SSH, hem de tenir en compte que es propaguen les variables d'entorn de l'usuari remot. Això fa que LC* es configuri a per exemple el català i que al demanar el manual de documentació d'algunes pàgines falli. A més l'idioma del sistema ja no serà consistent.

Per resoldre-ho hem de modificar /etc/ssh/sshd_config i editar la variable AcceptEnv comentant les línies que no vulguem permetre que es passin per l'entorn.

La configuració de Denyhosts és l'habitual de CIMNE, es modifica /etc/denyhosts.conf amb els següents paràmetres auto-explicats al fitxer de configuració:

```
DENY_THRESHOLD_ROOT = 3
DENY_THRESHOLD_RESTRICTED = 1
WORK_DIR = /var/lib/denyhosts
SUSPICIOUS_LOGIN_REPORT_ALLOWED_HOSTS=YES
LOCK_FILE = /var/lock/subsys/denyhosts
SMTP_HOST = correu.cimne.upc.edu
SMTP_PORT = 24
SMTP_FROM = DenyHosts Cluster Acuario <reports@cimne.upc.edu>
SMTP_USERNAME=reports-at-cimne.upc.edu
SMTP_PASSWORD=******
SYSLOG_REPORT=YES
ALLOWED_HOSTS_HOSTNAME_LOOKUP=NO
AGE_RESET_VALID=5d
AGE_RESET_ROOT=25d
AGE_RESET_RESTRICTED=25d
AGE_RESET_INVALID=10d
```

5.2.3.3 Política de contrasenyes

Seguint la política de contrasenyes de CIMNE hem de definir que els usuaris hagin de definir-les amb almenys 8 caràcters i 3 tipus diferents de caràcters.

Modificarem /etc/pam.d/system-auth-ac deixant la línia password com segueix:

```
password requisite pam_cracklib.so try_first_pass retry=3 minlen=8 difok=3
```

Podem trobar el procediment detallat a la documentació de RHEL6 [47].

5.2.3.4 Còpies de seguretat

Crearem un directori a /mnt/cluster-data/backups on hi emmagatzemarem els backups interns realitzats. Hi guardarem també els logs d'aquests backups.

Dins /mnt/cluster-data/sbin crearem l'script “backup-system.sh” que s'encarregarà d'agafar els directoris importants definits dins /mnt/cluster-data/sbin/includefile i emmagatzemar-los amb rsync dins el directori anteriorment citat.

L'script /mnt/cluster-data/sbin/backup-system.sh és aquest:

```
#!/bin/sh
#
#Aquest script realitza un backup incremental de tot el sistema
mitjançant
#rsync.
#
# Felip Moll 19/07/2012
#
ORIGEN=/
DESTI=/data0/backups/rsync
LOGDIR=/data0/backups/logs/rsync
EXCLUDEFILE=/data0/sbin/excludefile
INCLUDEFILE=/data0/sbin/includefile

if [ ! -x $DESTI ]; then mkdir -p $DESTI; fi
if [ ! -x $LOGDIR ]; then mkdir -p $LOGDIR; fi

cd $LOGDIR
find *.log -mtime +10 -exec rm {} \;

rsync -ar --delete --log-file=$LOGDIR/`date +%A\,%d-%m-%y`.log
--exclude-from=$EXCLUDEFILE --files-from=$INCLUDEFILE $ORIGEN $DESTI
```

Per altre banda la llista de fitxers inclosos definits a /mnt/cluster-data/sbin/excludefile és aquesta:

```
#Directoris i fitxers dels que fer backup
#
# Felip Moll 18-07-2012
/root
/etc
/var/lib/ganglia
/var/www
/opt/dell
```

Per executar periòdicament la còpia de seguretat crearem un crontab:

```
0 05 * * 1,3,5,7 /mnt/cluster-data/sbin/backup-system.sh
```

I finalment integrarem el clúster amb el sistema de backups de CIMNE instal·lant amb YUM el client de bàcula “bacula-fd” i configurant /etc/bacula-fd.conf amb els paràmetres que ens especifiquin (usuari i password del servidor de backups).

Finalment crearem un enllaç simbòlic de /data0 a /mnt/cluster-data ja que el sistema de CIMNE està configurat per realitzar còpies senceres i periòdiques dels directoris anomenats /data0.

5.2.3.5 Limitació de recursos per usuari

El límit que imposarem als usuaris és principalment el de memòria RAM al node màster. D'aquesta manera ens assegurarem que en el moment de compilar o visualitzar resultats en aquest node no puguin causar cap mal als processos principals del sistema i del gestor de cues.

La modificació es realitzarà a /etc/security/limits.conf

```
#<domain>        <type>  <item>         <value>
*                hard    memlock         unlimited
*                soft    memlock         unlimited
*                hard    as              6002779
slurm            hard    as              unlimited
root             hard    as              unlimited
```

La quantitat de memòria física real assignada a un usuari qualsevol (as) haurà de ser calculada en relació a l'ús que en fan els usuaris. Voldrem assegurar 2GiB de RAM als processos del sistema i al gestor de cues:

Si tenim 50 usuaris i un total de 32GB de RAM, 30 GiB seran disponibles per aquests usuaris i equitativament correspondria a 30/50 = 600MiB/usuari. No obstant aquesta mesura és massa restrictiva perquè en tot moment no hi ha tots els usuaris funcionant al màxim, per tant si suposem un 10% dels usuaris actius a ple rendiment i fem el càlcul correspondria a 6GiB per usuari.

En principi col·loquem el límit de 6GiB que en kB és 6002779kB, i en un futur si es desitja es podrà refinar aquest paràmetre.

En moments posteriors també es podrà analitzar l'ús de “cgroups” de Linux per tal d'afinar més el control de recursos per usuari i grup. Fins el moment la mesura aplicada és suficient.

5.2.3.6 Dell OpenManage OMSA

Podem trobar la informació i el software a la pàgina de suport de Dell dirigint-nos a l'apartat *Drivers and Downloads* un cop posat el Service Tag del node màster (A.3.2 Service Tags).

El paquet que hem de localitzar i descarregar-nos és:

Dell OpenManage Server Administrator Managed Node (Distribution Specific)

OM-SrvAdmin-Dell-Web-LX-6.5.0-2247.RHEL6.x86_64_A01.5.tar.gz

Llavors seguirem les instruccions definides al fitxer:

Custom Instructions for OM-SrvAdmin-Dell-Web-LX-6.5.0-2247.RHEL6.x86_64_A01.5.tar.gz

En cas de que no puguem instal·lar el paquet per problemes amb la versió de sistema operatiu no detectat (SL vs RHEL), podem instal·lar manualment els paquets RPM o col·locar-los al repositori /mnt/cluster-data/repo/cluster-packages. Aquesta última alternativa és la que escollirem. Col·locarem tots els paquets RPM a /mnt/cluster-data/repo/cluster-packages/OMSA/ .

Una forma alternativa de instal·lar-lo és mitjançant el repositori de Dell de software per Linux. Podem trobar les instruccions a:

http://linux.dell.com/repo/hardware/OMSA_6.5.1/

Un cop instal·lat OMSA es pot procedir a actualitzar el firmware del chassis i de la intefície iDRAC nodes.

5.2.3.7 Dell CMC & iDrac

L'accés al Dell CMC es realitza mitjançant una interfície web. Hem de configurar primer de tot la IP del dispositiu que segons l'esquema determinat és 172.26.0.200 (veure A.3.4 Adreçament IP). Ho farem des del Mòdul LCD del chassis. El procés és intuïtiu i no el detallarem aquí.

L'usuari estàndard d'accés i la seva contrasenya són:

```
Usuari:    root
Password:  calvin
```

Una vegada haguem accedit a la web veurem l'estat del clúster i podrem modificar les adreces IP relatives a l'iDrac de tots els altres nodes. També canviarem la contrasenya a la del nivell adequat segons el departament de Sistemes.

5.2.3.8 Dell PowerVault MD3000i

Instal·lació de Dell MD Storage Manager

Per la configuració de la cabina de discs necessitem el software proporcionat per Dell, "*MD Storage Manager*". La versió necessària és la corresponent a la MD3000i i només es suportada per Redhat 5, Suse 10 i anteriors.

Per tal de disposar d'aquest software hem de descarregar el Resource Manager CD de la MD3000i de Dell, amb nom *md3000i_2_2_0_18_R294936.iso :*

http://www.dell.com/support/drivers/us/en/04/DriverDetails?driverId=R294936&fileid=2731116446

Ens assegurarem que disposem de les biblioteques de 32 bits de X Windows instal·lades al sistema sota /usr/lib. En cas que en falti alguna s'instal·larà amb yum.

- libX11.so
- libXau.so
- libXdmcp.so
- libXext.so
- libXft.so
- libXi.so
- libXrender.so
- libXt.so
- libXtst.so

Llavors muntarem la imatge iso que hem descarregat per exemple a /media/.

Si executem l'instal·lador que es troba a /media/linux/install.sh quan provem de seleccionar qualsevol opció no trobarà una versió del SO compatible i el programa avortarà.

Per solucionar-ho instal·larem manualment el MD3000i Storage Manager.

Obviarem la instal·lació del Multi-pathing Driver ja que no és compatible amb kernels superiors al 2.6.18 i només serveix quan disposem de connexió SAS amb la cabina. Podem trobar més informació a [48].

El Modular Disk Configuration Utility tampoc l'instal·larem ja que només és un assistent i es poden realitzar les tasques des de l'Storage Manager.

1. Accedirem a /media/linux/app/ i executarem *SMIA-LINUX-03.35.A6.58.bin* . Apareixerà un assistent gràfic (Figura 67) on només haurem de seguir els passos.

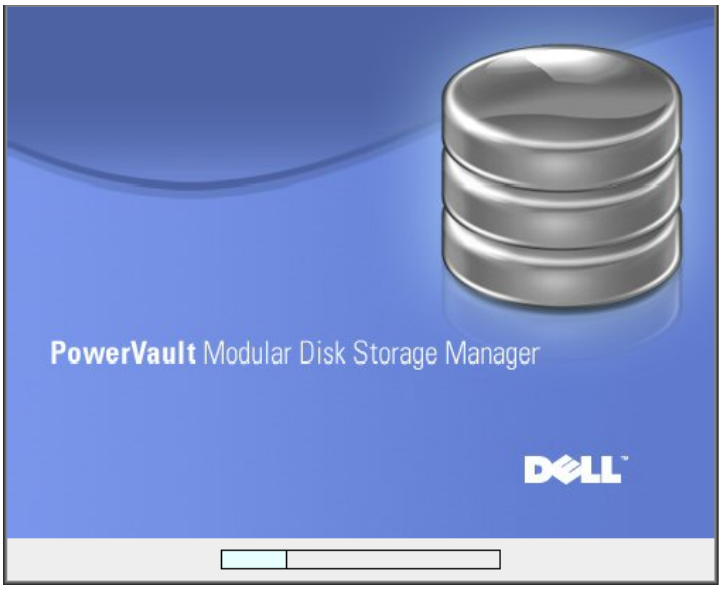


Figura 67: Inici instal lació MD Storage Manager

Escollirem la instal·lació típica per instal·lar tot el software, Figura 68.

Figura 68: Escollint tipus d'instal lació MD Storage Manager

El directori d'instal·lació que escollirem serà /opt/dell/mdstoragemanager, Figura 69.

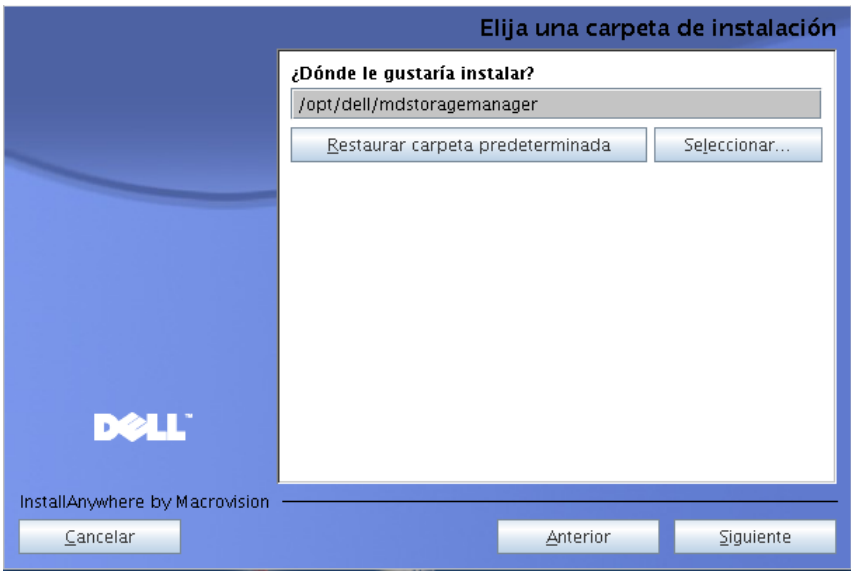


Figura 69: Directori destí de MD Storage Manager

3. Una vegada acabat l'assistent comprovarem que la instal·lació funciona accedint a /opt/dell/mdstoragemanager i executant:

```
[root@acuario mdstoragemanager]# ./client/SMclient
```

És possible que doni incompatibilitats amb la màquina virtual de Java. Si fos el cas només hi ha que substituir el directori /opt/dell/mdstoragemanager/jre amb una instal·lació d'una màquina de Java actualitzada.

Per fer més comoda l'execució crearem un enllaç simbòlic de */opt/dell/mdstoragemanager* a *client/SMclient .*

4. Un cop acabada la instal·lació podrem actualitzar el firmware dels components amb els binaris instal·lats (p.ex. Smfwupgrade). No actualitzarem el component NVRAM.

Configuració dels discs virtuals

L'únic que faltarà ara és configurar els discs virtuals al SAN. El procés és intuïtiu i no cal entrar en detall.

Només mencionem que crearem dos discs virtuals, un per /home i un per /mnt/cluster-data. Cada un d'aquests discs anirà lligat a un conjunt de discs físics. Especialment crearem un raid 5 per /home i un raid 1 per /mnt/cluster-data.

Configurarem també la IP de out-band-management (Raid Controller Module 0, només tenim un mòdul de gestió) que farem servir i que serà la 172.26.0.201 segons hem especificat a Adreçament.

Configurarem les dues adreces dels ports iSCSI 0 i 1 amb les IP 10.50.0.200 i 10.50.0.201.

Finalment haurem de donar accés a Acuario als volums creats. Per fer-ho comprovarem quin target iSCSI identifica Acuario i l'afegirem a la llista de hosts permesos en aquests volums concrets Figura 70 i Figura 71:

```
[root@acuario mdstoragemanager]# iscsi-iname
iqn.1994-05.com.redhat:fdc5ec71f53d
```

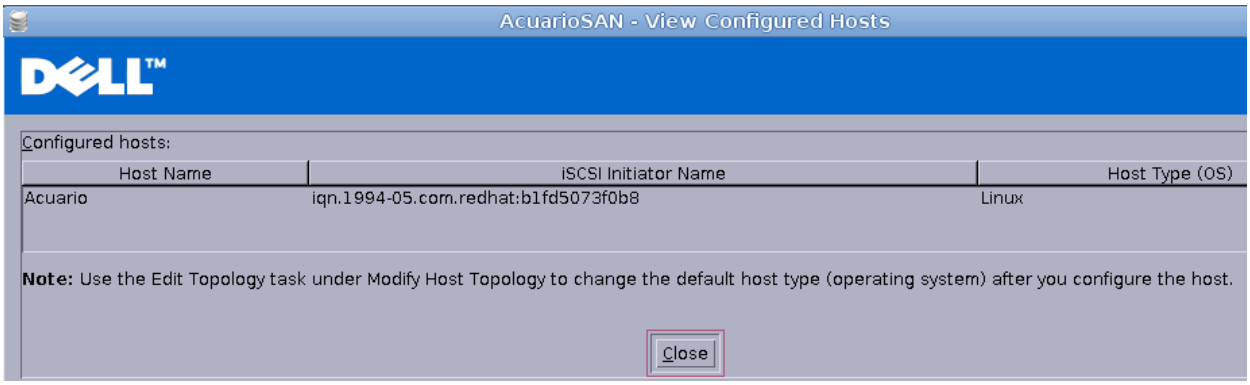


Figura 70: Host Acuario amb l'identificador iSCSI configurat al SAN

Podem trobar dos videos de configuració realitzats en el moment de re-estructuració de la cabina. Els videos es troben a l'Annex 1.1 de deocuments interns, referències [doc16] i [doc17].

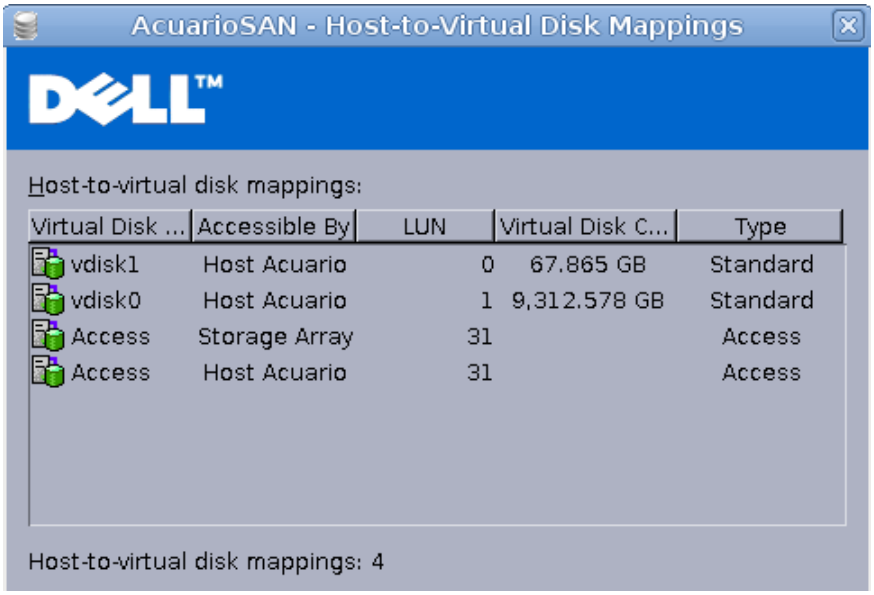


Figura 71: Enllaç de host Acuario a discs virtuals vdisk1 i vdisk0.

L'aspecte general de la pantalla de MD Storage Manager el veiem a la Figura 72.

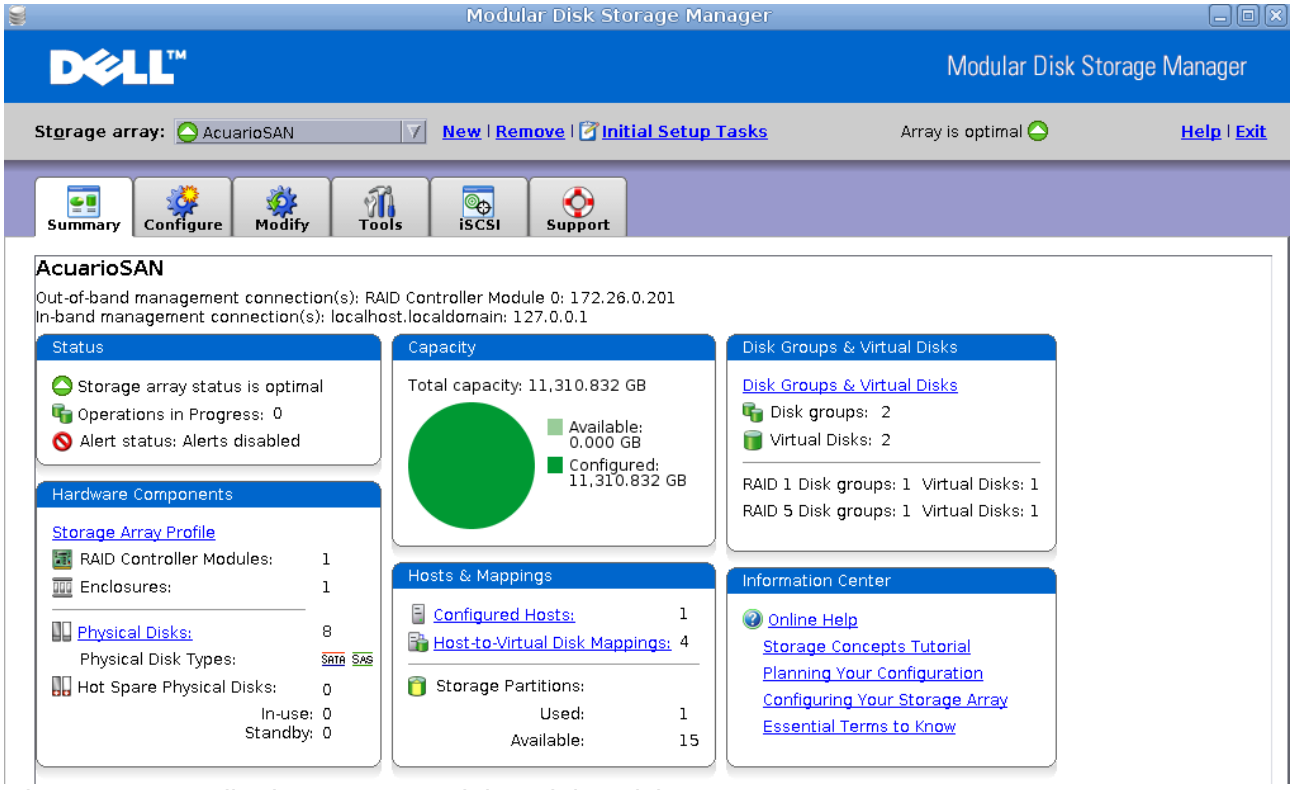


Figura 72: Pantalla de "summary" del Modular Disk Storage Manager

5.2.4 Capa 3 – Serveis de clustering

5.2.4.1 Configuració de NFS

Instal·larem el servei NFS amb YUM i l'activarem amb *chkconfig nfs on.*

Llavors configurarem el servei de la següent manera:

```
[root@acuario ~]# cat /etc/exports
/repo       10.0.1.0/24 (ro)
/kickstart  10.0.1.0/24 (ro,no_root_squash)
/globalfs   10.0.1.0/24 (sync,no_wdelay,subtree_check,rw,no_root_squash)
/home       10.0.1.0/24 (sync,no_wdelay,subtree_check,rw,no_root_squash)
```

Veiem com totes les connexions les feim mitjançant la xarxa ethernet. Això és així ja que encara que per Infiniband disposaríem d'una latència menor, la comunicació entre el node màster i el SAN és per ethernet i per tant és en aquest punt on hi ha el coll d'ampolla.

És important veure el paràmetre “sync” que permet que els canvis siguin escrits a disc instantàniament. El cas de “async” és més segur però menys ràpid i els usuaris noten que no reben la sortida dels seus fitxers en temps real.

5.2.4.2 Repositori de software YUM

Crearem el directori /mnt/cluster-data/repo/ on hi emmagatzemarem els diferents repositoris que ens interessarà mantenir.

El principal repositori serà l'oficial de Scientific Linux. A la seva pàgina web expliquen com crear un repositori rèplica dels seus [49]. Bàsicament consisteix en una comanda rsync que s'executa de forma periòdica i que connectant amb els seus servidors manté sincronitzada la còpia local.

Crearem per tant el directori /mnt/cluster-data/repo/scientific/<versió>/<arquit.>/ i un script que situarem a /mnt/cluster-data/sbin/update-repositoris.sh. L'script s'executarà periòdicament amb cron:

/mnt/cluster-data/sbin/update-repositoris.sh:

```
#!/bin/bash
# Aquest script actualitza el repositori local.

OPTS="-avlH –delete --exclude=iso –exclude=archive/obsolete    \
--exclude=updates/fastbugs –exclude=sites/Fermi –exclude=debug \
--exclude=repoview --exclude=headers –exclude=*debuginfo*"     \
REMOTE_MIRROR="rsync://rsync.scientificlinux.org/scientific/6.1/x86_64/"
LOCAL_MIRROR="/mnt/cluster-data/repo/scientific/6.1/x86_64/"

rsync $OPTS $REMOTE_MIRROR $LOCAL_MIRROR
```

Un segon repositori que ens interessarà afegir serà el de EPEL que disposarà de software secundari i actualitzat. Crearem el directori /mnt/cluster-data/repo/epel/<versió>/<arquit.> i afegirem per tant a l'script:

```
REMOTE_MIRROR="rsync://mirror.nexicom.net/Fedora-EPEL/6/x86_64/"
LOCAL_MIRROR="/mnt/cluster-data/repo/epel/6/x86_64/"
OPTS="-avlH --delete --exclude=iso --exclude=debug \
--exclude=repoview --exclude=archive/obsolete"

rsync $OPTS $REMOTE_MIRROR $LOCAL_MIRROR
```

El crontab vindrà definit com segueix:

```
30 03 * * * /mnt/cluster-data/sbin/update-repositoris.sh 2>&1 > \
/var/log/update-repositoris.log
```

Per altra part voldrem crear el nostre propi repositori de paquets personalitzats. Ho farem creant el directori /mnt/cluster-data/repo/cluster-packages i situant allà dins tots els RPM directament o en subdirectoris i entrant al directori executarem:

```
[root@acuario x86_64]# cd /mnt/cluster-data/repo/cluster-packages/x86_64/
[root@acuario x86_64]# createrepo .
```

Si volem afegir en el futur paquets rpm al repositori només els hi haurem de copiar i entrant al directori executar:

```
[root@acuario x86_64]# createrepo –-update .
```

El punt final que resta configurar és el de definir els fonts de Yum. Accedirem a /etc/yum.repos.d/ i crearem el fitxer cluster-packages.repo:

```
[root@acuario yum.repos.d]# cat cluster-packages.repo
[cluster-packages]
name=Paquets del cluster Acuario personalitzats
baseurl=file:///repo/cluster-packages/x86_64/
enabled=1
priority=1
gpgcheck=0
```

Modificarem també el fitxer sl-other.repo i desactivarem posant enabled=0 tots els fonts ja que són dedicats a desenvolupament de codi o proves.

Modificarem sl.repo, fitxer de fonts oficials i desactivarem el de fitxers SRPM. Al [sl] i [sl-security] els hi comentarem el baseurl i hi especificarem un baseurl nou:

Per [sl]:

```
baseurl=file:///repo/scientific/$releasever/$basearch/os/
```

i per [sl-security]:

```
baseurl=file:///repo/scientific/$releasever/$basearch/updates/security/
```

Finalment per facilitar l'accés al repositori crearem un enllaç simbòlic de /repo a /mnt/cluster-data/repo.

Provarem els repositoris amb:

```
yum clean all
yum update
```

5.2.4.3 Network Time Protocol

És important disposar d'un servei de NTP funcionant per tal de sincronitzar totes les dates dels diferents nodes del clúster. De no fer-ho podríem ocasionar problemes greus en el gestor de treballs i els monitors del clúster.

L'únic servidor que té accés directe a Internet és el node màster, per tant aquest s'encarregarà de sincronitzar la data amb els servidors d'Internet i proporcionar el servei NTP als nodes restants.

Configurem Acuario per tal que agafi l'hora dels servidors d'Internet i ajusti la del propi hardware (no seria necessari si a la instal·lació haguessim marcat que s'utilitzés NTP):

```
yum install ntp ntpdate
chkconfig ntpdate on
ntpdate -u -b -s server 0.rhel.pool.ntp.org
hwclock --utc --systohc
```

Modifiquem també /etc/ntp.conf per tal de restringir els servidors que poden demanar-nos la data. Concretament ens fixem en les següents línies:

```
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1
# Hosts on local network are less restricted.
restrict 10.0.1.0 mask 255.255.255.0 nomodify notrap
```

Comprovem al mateix fitxer que hi hagi els servidors de RedHat com a servidors de ntp:

```
# Use public servers from the pool.ntp.org project.
server 0.rhel.pool.ntp.org
server 1.rhel.pool.ntp.org
server 2.rhel.pool.ntp.org
```

Finalment executem el servidor NTPD:

```
chkconfig ntpd on
service ntpd start
```

5.2.4.4 Definició dels hosts

Haurem de definir a /etc/hosts quins són els nodes dels que disposarà el clúster, tant per facilitar l'accés a ells per ssh (només root), com per fer pings, realitzar configuracions del sistema i com a requisit per la pila de serveis de clúster.

El fitxer seguirà un format estricte ja que aprofitarem per fer-lo servir en el desplegament de nodes amb TFTPBoot.

El fitxer tindrà el següent cos:

```
[root@acuario .ssh]# cat /etc/hosts
#Atenció, el format d'aquest fitxer ha de ser obligatoriament el seguent:
#
#<ip_ib_node><tabulador><pezXXX><espai><ib_pezXXX><enter>
#<ip_eth_node><tabulador><eth_pezXXX>
#
# No respectar aquests formats de línies, farà que no es configuri bé el nom
# de host al reiniciar el node, després de la primera instalació.
# (Veure ks.cfg).
#
127.0.0.1   localhost.localdomain   localhost
::1         localhost6.localdomain6 localhost6
147.83.143.111    acuario.cimne.upc.edu
10.0.0.100  ib_acuario
10.0.1.100  eth_acuario acuario
10.0.0.1    pez001 ib_pez001
10.0.0.2    pez002 ib_pez002
10.0.0.3    pez003 ib_pez003
10.0.0.4    pez004 ib_pez004
10.0.0.5    pez005 ib_pez005
10.0.0.6    pez006 ib_pez006
10.0.0.7    pez007 ib_pez007
10.0.0.8    pez008 ib_pez008
10.0.0.9    pez009 ib_pez009
10.0.0.10   pez010 ib_pez010
10.0.0.11   pez011 ib_pez011
10.0.0.12   pez012 ib_pez012
10.0.0.13   pez013 ib_pez013
10.0.0.14   pez014 ib_pez014
10.0.0.15   pez015 ib_pez015
10.0.1.1    eth_pez001
10.0.1.2    eth_pez002
10.0.1.3    eth_pez003
10.0.1.4    eth_pez004
10.0.1.5    eth_pez005
10.0.1.6    eth_pez006
10.0.1.7    eth_pez007
10.0.1.8    eth_pez008
10.0.1.9    eth_pez009
10.0.1.10   eth_pez010
10.0.1.11   eth_pez011
10.0.1.12   eth_pez012
10.0.1.13   eth_pez013
10.0.1.14   eth_pez014
10.0.1.15   eth_pez015
```

5.2.4.5 Sincronització d'usuaris, grups i hosts

Per tal de tenir sincronitzats tots els usuaris, contrasenyes, grups i noms de host entre el node màster i la resta de nodes, serà necessari instal·lar el servei NIS. L'autenticació del sistema es modifica a PAM donant com a opció d'autenticació el client NIS.

Part de servidor

El servidor NIS que instal·larem serà Ypserv.

1. Instal·lar ypserv

```
yum install ypserv
```

2. Indicar el NIS DOMAIN NAME

```
nisdomainname cimne.upc.edu
```

3. Editar el fitxer /etc/sysconfig/network afegint al final:

```
NISDOMAIN=cimne.upc.edu
```

4. Reiniciem la xarxa

```
service network restart
```

5. Ens assegurem que portmapper esta funcionant mitjançant:

```
rpcinfo -p localhost
```

6. Copiem el fitxer de configuració del servidor i l'editem afegint la xarxa a la que volem que doni servei (255.255.255.0 10.0.1.0):

```
[root@acuari ~]# cp /usr/share/doc/ypserv-2.19/securenets  /var/yp/
[root@acuari ~]# tail /var/yp/securenets
# Always allow access for localhost
255.0.0.0  127.0.0.0
# Allow acces to LAN
255.255.255.0   10.0.1.0
```

7. Iniciar serveis i marcarlos per l'inici:

```
service yppasswdd start
service ypserv start
chkconfig yppasswdd on
chkconfig ypserv on
```

8. Executem el següent, afegint només eth_acuari (per defecte l'agafa automàticament, és a dir, podem polsar directament ctrl+d):

```
/usr/lib/yp/ypinit -m
```

Part dels clients

El procés de configuració dels nodes es realitzarà de forma automàtica des de l'script de Kickstart. Tot i així s'explica aquí la forma manual de configuració:

1. Configurar el NIS Domain Name:

```
nisdomainname cimne.upc.edu
```

2. Editar el fitxer /etc/sysconfig/network afegint:

```
NISDOMAIN=cimne.upc.edu
```

3. Reiniciem la xarxa:

```
service network restart
```

4. Editem el fitxer /etc/yp.conf i afegim:

```
domain cimne.upc.edu server 10.0.1.100
```

5. Iniciam el servei:

```
service ypbind start
chkconfig ypbind on
```

6. Comprovem que està funcionant el servei RPC:

```
rpcinfo -u localhost ypbind
```

7. Modifiquem /etc/host.conf afegint la paraula "nis":

```
order hosts, nis, bind
```

8. Editem el fitxer /etc/nsswitch.conf (no passa res si hi ha més mecanismes que “files” i “nis”):

```
passwd: files nis
shadow: files nis
group: files nis
hosts: files dns
netgroup: files nis
automount: files nis
aliases: files nis
```

9. Comprovem que un getent passwd ens mostra els mateixos usuaris que els que hi ha a Acuari.

Gestió amb NIS

Per crear/eliminar un usuari es fa de la forma habitual, executant posteriorment l'ordre make de YP i reiniciant el servei.

Per exemple:

```
[root@acuari ~]# useradd nomusuarinou
[root@acuari ~]# passwd nomusuarinou
[root@acuari ~]# cd /var/yp/
[root@acuari yp ]# make
[root@acuari yp ]# service ypserv reload
```

Per evitar el problema que podria sorgir si un administrador s'oblidés d'executar aquestes comandes crearem un script que s'executarà cada vegada que es crei o modifiqui un usuari o un grup. A més instal·larem un cron que executarà l'script de forma periòdica:

```
[root@acuario .ssh]# cat /mnt/cluster-data/sbin/update-nis.sh
#!/bin/bash
# Script per actualitzar la BBDD de NIS quan s'actualitza un password, un
usuari o un grup.

if [ ! -f /etc/shadow.last ]; then cp /etc/shadow /etc/shadow.last; fi
if [ ! -f /etc/passwd.last ]; then cp /etc/passwd /etc/passwd.last; fi
if [ ! -f /etc/group.last ]; then cp /etc/group /etc/group.last; fi

PWD_CHANGED=`diff /etc/passwd /etc/passwd.last`
SHW_CHANGED=`diff /etc/shadow /etc/shadow.last`
GRP_CHANGED=`diff /etc/group /etc/group.last`

if [ "$PWD_CHANGED" != "" ]; then echo "/etc/passwd -> $PWD_CHANGED";
CH="TRUE"; cp -p /etc/passwd /etc/passwd.last; fi
if [ "$SHW_CHANGED" != "" ]; then echo "/etc/shadow -> Canviat.."; CH="TRUE";
cp -p /etc/shadow /etc/shadow.last; fi
if [ "$GRP_CHANGED" != "" ]; then echo "/etc/group -> $GRP_CHANGED";
CH="TRUE"; cp -p /etc/group /etc/group.last; fi

if [ "$CH" != "" ]
then
    date
    cd /var/yp
    make
    service ypserv restart
    echo "----------------------------------------------"

    #Només en cas de tenir la opció "CacheGropus=1" a /etc/slurm/slurm.conf,
    #i que s'hagi modificat algun grup o el passwd del sistema hem de fer
    #scontrol reconfig
    scontrol reconfig
fi
```

Crontab:

```
0 0 * * * /mnt/cluster-data/sbin/update-nis.sh 2>&1 \
>> /var/log/update-nis.log
```

5.2.4.6 SLURM

Requisits previs:

- Disposar de tots els serveis de la capa 3 explicats prèviament instal·lats i configurats.

1. Generarem unes claus SSH per copiar-les posteriorment als nodes amb Kickstart i permetre que el servidor màster Acuario (usuari root) s'hi pugui connectar sense necessitat d'introduir contrasenya. Aquest pas és necessari per el gestor de recursos que necessitarà accedir als nodes.

Executem les següents ordres com a super-usuari:

```
ssh-keygen
cp /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys.acuari o
chmod -R go= /root/.ssh
```

2. Instal·lació de Munge, des de els repositoris de EPEL:

```
yum update && yum install munge
```

Comprovarem els permissos dels directoris de Munge tal com ens recomanen a la seva pàgina web [53]. Han de ser tots propietat de l'usuari i grup munge.

- /etc/munge/ Conté la clau privada del dimoni, els permisos han d'estar a 0700.
- /var/lib/munge/

Conté una llavor per el generador de nombres aleatoris i manté informació del procés i les autenticacions. Els permissos recomanats són 0711.

- /var/log/munge/ Logs del dimoni, permisos a 0700.
- /var/run/munge/ Sockets i pid, permisos 0755.

Crearem una clau privada que es situarà a /etc/munge/munge.key

```
dd if=/dev/random bs=1 count=1024 >/etc/munge/munge.key
```

Configurarem l'inici de Munge i iniciarem el servei:

```
chkconfig munge on
service munge start
```

Comprovarem que funciona:

```
[fmoll@acuario ~]$ remunge
2012-10-05 18:22:08 Spawning 1 thread for encoding
2012-10-05 18:22:08 Processing credentials for 1 second
2012-10-05  18:22:09  Processed  12475  credentials  in  1.000s  (12472
creds/sec
```

3. Descarregar SLURM des de la pàgina web de Sched MD. Llavors desempaquetar-lo i modificar slurm.spec per incloure suport per MySQL.

   A continuació empaquetar-lo de nou i crear un RPM.

```
[root@acuario]# yum install readline-devel openssl-devel munge-devel pam-devel gtk2.x86_64 gtk2-devel.x86_64

[root@acuario]# wget
 http://www.schedmd.com/download/latest/slurm-2.x.tar.bz2

[root@acuario]# tar xvjf slurm-2.x.tar.bz2
```

Editem slurm.spec, eliminant la opció de no infloure MySQL i afegint l'opció

```
%slurm_with_opt mysql
```

Empaquetem de nou i creem l'RPM.

```
rpmbuild -ta slurm-2.x.tar.bz2
```

Moure els RPM generats al repositori personalitzat:

```
mv /root/rpmbuild/RPMS/ /repo/cluster-packages/x86_64/
cd /repo/cluster-packages/x86_64/
createrepo --update .
```

4. Instal·lar Slurm al node màster

   yum install slurm*

5. Als nodes esclau només cal instal·lar: slurm, slurm-munge i slurm-plugins.
6. Configurar /etc/slurm.conf i /etc/slurmdbd.conf al node màster.
7. Els nodes esclau utilitzen el mateix fitxer de configuració que el màster, per tant farem que tots els nodes, màster inclòs, tinguin un enllaç simbòlic de /etc/slurm/slurm.conf a /mnt/cluster-data/globalfs/etc/slurm/slurm.conf.

   Els fitxers de configuració es poden trobar als annexos A.4.4 /etc/slurmdbd.conf i A.4.5 /etc/slurm/slurm.conf - /globalfs/etc/slurm.conf.
8. Instal·lar MySQL i configurar-lo seguint els senzills passos definits a la documentació de SLURM, apartat “Accounting” [54].

   El procés passa per crear una base de dades anomenada *slurm_acct_db* i un usuari *slurm* amb permisos sobre aquesta.
9. Afegir slurmdbd a chkconfig amb *chkconfig --add slurmdbd* i iniciar el servei.

5.2.4.7 Kickstart + PXE

Requisits

- Tenir la xarxa configurada de la forma que s'explica a 5.2.4.4. Revisar dues vegades el contingut de /etc/hosts.
- Tenir l'estructura de directoris creada correctament, tal com s'explica a 5.1.6 incloent els enllaços simbòlics corresponents de /kickstart a /mnt/cluster-data/kickstart i /tftpboot a /mnt/cluster-data/tftpboot.
- Haver exportat els directoris per NFS tal com s'indica a 5.2.4.1.

Estructura de Kickstart + PXE Boot

Disposem dels serveis següents:

TFTPBoot: Servidor FTP molt senzill que apuntarà a un directori on hi haurà alguns fitxers per realitzar la càrrega inicial del sistema: un fitxer de configuració amb diverses opcions d'engegada, i un gestor d'engegada.

L'objectiu d'aquest servei és proporcionar tot això a les màquines que mitjançant el protocol PXE ho demanin.

Protocol PXE: Permetrà engegar per xarxa i farà que s'obtinguin els fitxers necessaris del servidor TFTP. Els carregarà en memòria mostrant a l'usuari un menú semblant al Lilo o al Grub i permetrà iniciar amb una imatge de kernel ubicada a una unitat NFS que també contindrà els fitxers de l'instal·lador de Scientific Linux.

NFS: Necessari per exportar els fitxers de l'instal·lador de Scientific Linux.

DHCP: Servei necessari per fer que les màquines que executin amb PXE Boot (Network Boot), puguin obtenir una adreça IP i començar la instal·lació de Scientific Linux, important els directoris NFS.

Kickstart: Servei que conté una configuració de YUM bàsica per tal d'automatitzar la instal·lador de sistemes basats en RedHat Linux.

A la Figura 73 mostrem un esquema del funcionament de tot el procés.

Instal·lació

Primer de tot, necessitem el CD ISO de S.L.6.0 per tal d'obtenir la imatge del kernel:

```
cd /repo/scientific/6.0/x86_64/iso/
wget   http://ftp1.scientificlinux.org/linux/scientific/6.0/x86_64/iso/SL-60-
x86_64-2011-03-03-Install-DVD.iso
mount -t iso9660 -o loop SL-60-x86_64-2011-03-03-Install-DVD.iso  /media/
```

Instal·lem els serveis basics:

```
yum install dhcp xinetd tftp tftp-server syslinux
```

A continuació creem l'estructura per el servidor TFTPBoott:

```
mkdir -p /tftpboot/pxelinux.cfg /tftpboot/images/scientific/6/x86_64/
```

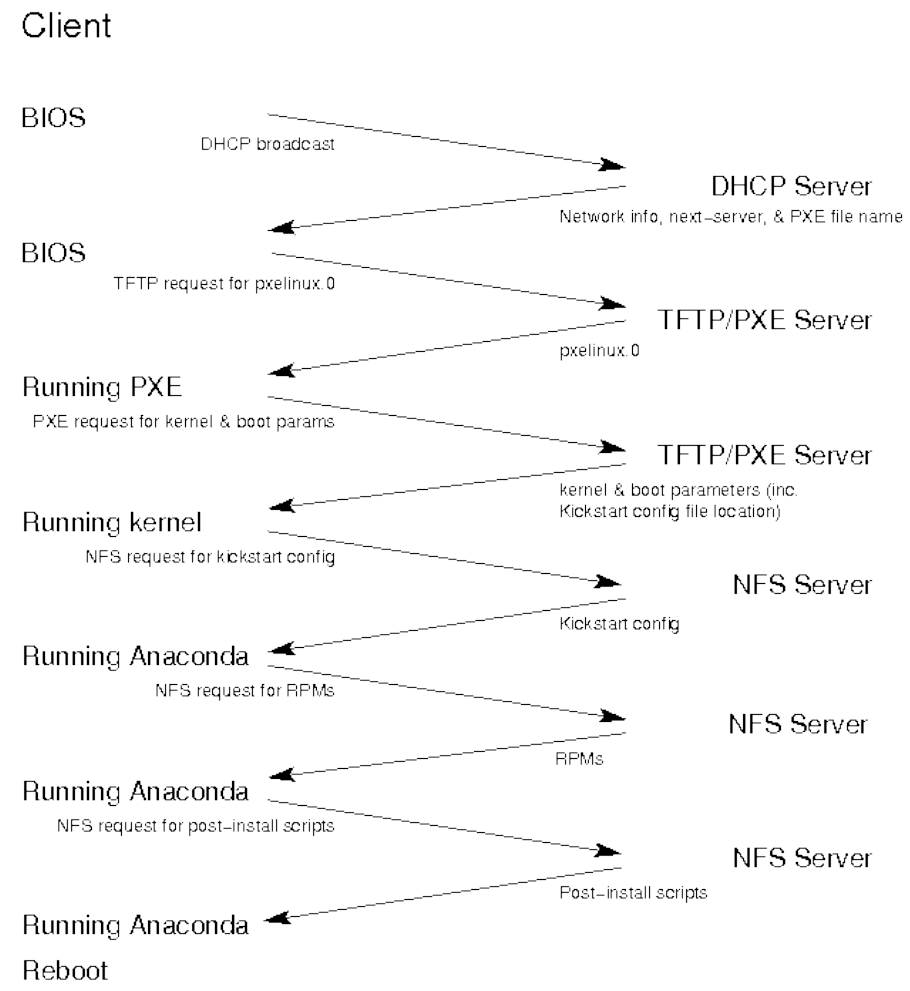


Figura 73: Procés PXE + Kickstart

Copiem el kernel i el ramdisk al directori del TFTPboot:

```
rsync -avP /media/isolinux/initrd.img /media/isolinux/vmlinuz    \
 /tftpboot/images/scientific/6/x86_64/
```

Copiem també els fitxers per executar el menú d'engegada:

```
cd /usr/share/syslinux/
rsync -avP chain.c32 mboot.c32 memdisk menu.c32 pxelinux.0 /tftpboot/
```

Creem l'estructura per Kickstart:

```
cd /kickstart
mkdir -p /kickstart/acuari/etc/
cd /kickstart/acuari/etc/
mkdir -p rc.d/init.d profile.d ssh yum/pluginconf.d
cp -Rpf /etc/yum.repos.d/ .
```

Ara ja tenim l'estructura del TFTPBoot i del Kickstart. A continuació passarem a configurar el menú d'engegada del TFTPboot.

Crearem el fitxer /tftpboot/pxelinux.cfg/default i l'omplirem per exemple de la següent manera (disponible també a l'annèx A.4.1 /tftpboot/pxelinux.cfg/default):

```
# Fitxer de configuració del menu de PXE BOOT
# Opcions rellevants:
# ONTIMEOUT : Indica quin LABEL triar passats TIMEOUT milisegons.
# LABEL : Defineix diferents opcions d'engegat
# "biosdevname=0" : En nodes Dell nous, la bios permet nombrar les
# interficies pel nom
# relacionat a la posició fisica real. Nosaltres volem el sistema classic
# perque potser no tots els nodes en el futur seran Dell.
DEFAULT menu.c32
PROMPT 0
TIMEOUT 100

#ONTIMEOUT scientific
ONTIMEOUT local

NOESCAPE 1
ALLOWOPTIONS 0
MENU TITLE Acuario PXE Menu

#Definicio del menú
LABEL local
MENU LABEL Inicia desde disc dur
LOCALBOOT 0

LABEL scientific
MENU LABEL Scientific 6.1 Instalacio Node
KERNEL images/scientific/6.1/x86_64/vmlinuz
APPEND      initrd=images/scientific/6.1/x86_64/initrd.img      biosdevname=0
ramdisk_size=100000 ksdevice=eth0 ip=dhcp ks=nfs:10.0.1.100:/kickstart/ks.cfg

LABEL rescue
MENU LABEL Scientific Linux 6.1 Rescue
KERNEL images/scientific/6.1/x86_64/vmlinuz
APPEND      initrd=images/scientific/6.1/x86_64/initrd.img      biosdevname=0
ramdisk_size=10000            text            ksdevice=eth0            rescue
ks=nfs:10.0.1.100:/kickstart/rescue.cfg
```

Hi ha moltes opcions configurables que podem consultar a la documentació del servei.

Com a exemple es pot posar una contrasenya a una de les entrades generant-lo amb la comanda:

```
sha1pass my_password
```

Llavors la línia que mostrem a continuació serviria per definir una contrasenya a una entrada del menú:

```
MENU PASSWD $4$2Eg$ptuj0QeBgQhgbGKV93dzN/CAZgs$
```

Arribats a aquest punt editem /etc/xinetd.d/tftp indicant la ruta per defecte de TFTP per tal de fer que a l'inici s'engegui el servei:

```
server_args        = -s /tftpboot
disable                 = no
```

També creem un ntp.conf per els nodes:

```
echo "server 10.0.1.100" > /kickstart/acuario/etc/ntp.conf
```

Finalment configurem el servidor DHCP.

Primer farem que no pugui escoltar més que a eth0, la xarxa de ethernet dels nodes, i farem que doni IP's del rang 10.0.1.0/24. També activarem el protocol bootp i finalment indicarem per cada node la MAC que s'ha d'associar amb la IP i el nom de host. En aquest exemple només posem 3 nodes a mode d'exemple.

```
[root@acuario /]# cat /etc/sysconfig/dhcpd
# Command line options here
DHCPDARGS="eth0"
```

Després editem la seva configuració a /etc/dhcp/dhcpd.conf

(disponible a l'annex A.4.2 /etc/dhcp/dhcpd.conf):

```
# dhcpd.conf
# Sample configuration file for ISC dhcpd
 option domain-name "cluster.cimne.upc.edu";
 option domain-name-servers 147.83.143.133;
 default-lease-time 21600;
 max-lease-time 43200;
 ddns-update-style none;
 authoritative;
 log-facility local7;
 subnet 10.0.1.0 netmask 255.255.255.0 {
        range 10.0.1.1 10.0.1.15;
        option routers 10.0.1.100;
        option subnet-mask 255.255.255.0;
        option ntp-servers 10.0.1.100;
        allow booting;
        allow bootp;
        next-server 10.0.0.100;
        filename "/pxelinux.0";
       host pez001 {
              hardware ethernet 00:1E:C9:CD:2A:DC;
          fixed-address 10.0.1.1;
       }

     ----8<----------

        host pez015 {
               hardware ethernet 5C:26:0A:FE:2C:4C;
         fixed-address 10.0.1.15;
        }
 }
```

A continuació copiarem i crearem alguns fitxers que seran sincronitzats a tots els nodes i adequats des de la instal·lació de Kickstart:

```
cp /root/.ssh/id_rsa.pub /kickstart/acuario/authorized_keys
cp /etc/hosts /kickstart/acuario/etc
cp /etc/yum.repos.d/*.repo /kickstart/acuario/etc
cp /etc/munge/munge.key /kickstart/acuario/etc/munge/munge.key
```

- Fitxers de NIS, explicats a 5.2.4.3:

```
host.conf
nsswitch.conf
yp.conf
```

- Configuració de NTP:

```
[root@acuario etc]# cat ntp.conf
server 10.0.1.100
```

- Fitxer sshd_config copiat de /etc/sshd/sshd_acuario amb el paràmetre:

```
AllowUsers root
```

- Fitxer /kickstart/acuario/etc/sysconfig/network amb el contingut:

```
NETWORKING=yes
HOSTNAME=xxx
```

- Fitxers de configuració de xarxa:

```
[root@acuario etc]# cat sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
NM_CONTROLLED=yes
ONBOOT=yes
IPADDR=10.0.1.xxx
BOOTPROTO=none
NETMASK=255.255.255.0
TYPE=Ethernet
IPV6INIT=no
USERCTL=no
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
NAME="System eth0"

[root@acuario etc]# cat sysconfig/network-scripts/ifcfg-ib0
DEVICE=ib0
NM_CONTROLLED=yes
ONBOOT=yes
IPADDR=10.0.0.xxx
BOOTPROTO=none
NETMASK=255.255.255.0

[root@acuario etc]# cat sysconfig/network-scripts/ifcfg-ib1
NAME="System ib0"
TYPE=Ethernet
DEVICE="ib1"
ONBOOT="no"
```

Com a últim pas:

```
service xinetd restart
chkconfig dhcpd on
service dhcpd restart
```

Configuració de Kickstart

Com hem vist a la configuració del menú de PXE, un cop seleccionada la opció d'engegat es llegeix un fitxer de configuració de Kickstart per iniciar l'instal·lador.

En el nostre cas tenim les opcions:

```
LABEL scientific
MENU LABEL Scientific 6.1 Instalacio Node
...
ks=nfs:10.0.1.100:/kickstart/ks.cfg

LABEL rescue
MENU LABEL Scientific Linux 6.1 Rescue
...
ks=nfs:10.0.1.100:/kickstart/rescue.cfg
```

Editarem llavors aquests dos fitxers de configuració, ks.cfg i rescue.cfg.

Recomanem utilitzar l'eina “system-config-kickstart” per generar ks.cfg, Figura 74, Figura 75, Figura 76.

Configuració de rescat:

```
[root@acuari kickstart]# cat /kickstart/rescue.cfg
lang es_ES
keyboard es
mouse none
nfs --server=10.0.1.100 --dir=/repo/scientific/6.1/x86_64
network --bootproto=dhcp
```

Configuració de ks.cfg:

Al ser una configuració molt extensa s'adjunta a l'annex A.4.3 /kickstart/ks.cfg.

El fitxer es divideix en 3 parts:

Part 1: Paràmetres de l'instal·lador.

En aquesta part es defineixen els paràmetres tal com ho faríem si instal·léssim manualment el sistema; activació del firewall, directori d'instal·lació, particions, contrasenya de root, configuració de la xarxa, mode d'instal·lació, llengua, etc.

Part 2: %pre

Aquesta part defineix alguns scripts que es portaran a terme abans de començar a instal·lar paquets. Serveix bàsicament per muntar el repositori NFS.

Part 3: %post

Definició dels scripts que s'executaràn un cop acabada la instal·lació de paquets. Aquests scripts seran molt útils per fer configuracions automàtiques tals com editar l'fstab per tal de que es muntin les unitats NFS, activar l'Infiniband, actualitzar el sistema, copiar fitxers de configuració, configurar el rellotge del sistema, l'ssh, etc.

Part 4: %packages

Llista de grups de paquets (@grup) i de paquets individuals que s'han d'instal·lar o no instal·lar.

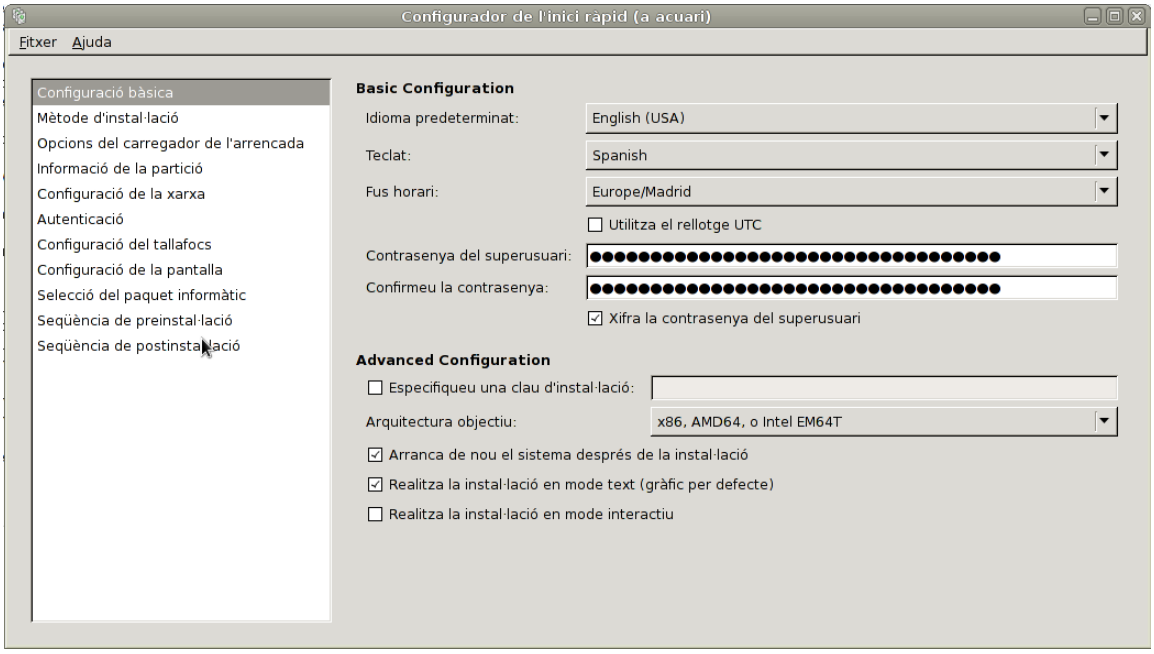


Figura 74: Configurador de Kickstart - Opcions bàsiques

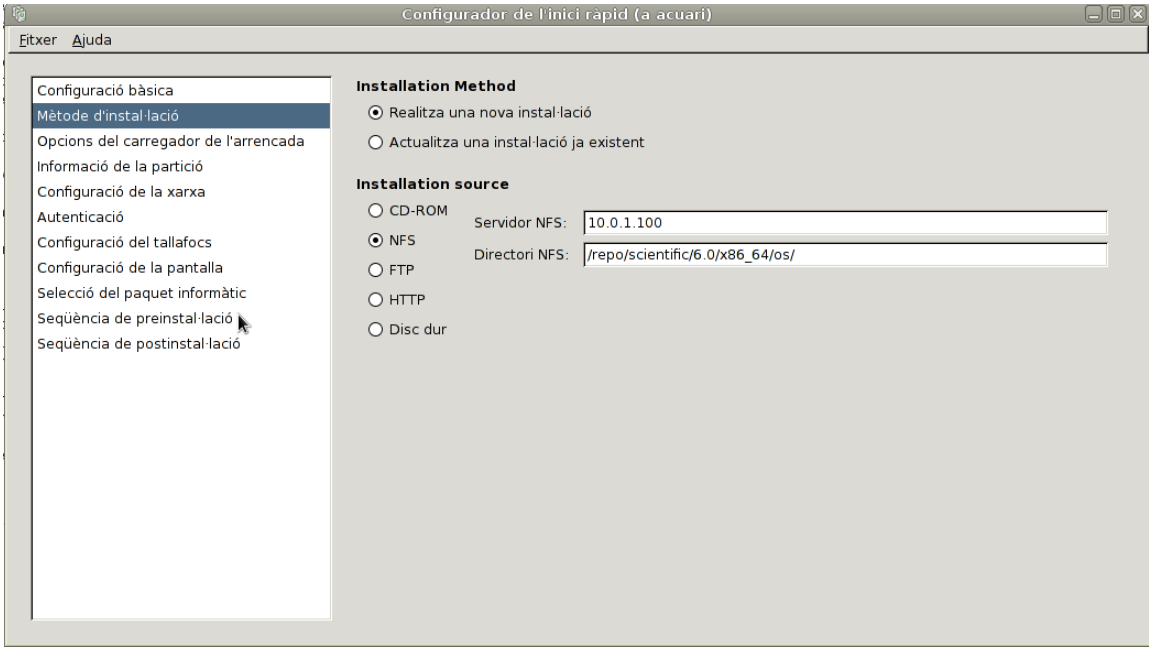


Figura 75: Configurador de Kickstart - Punt de muntatge

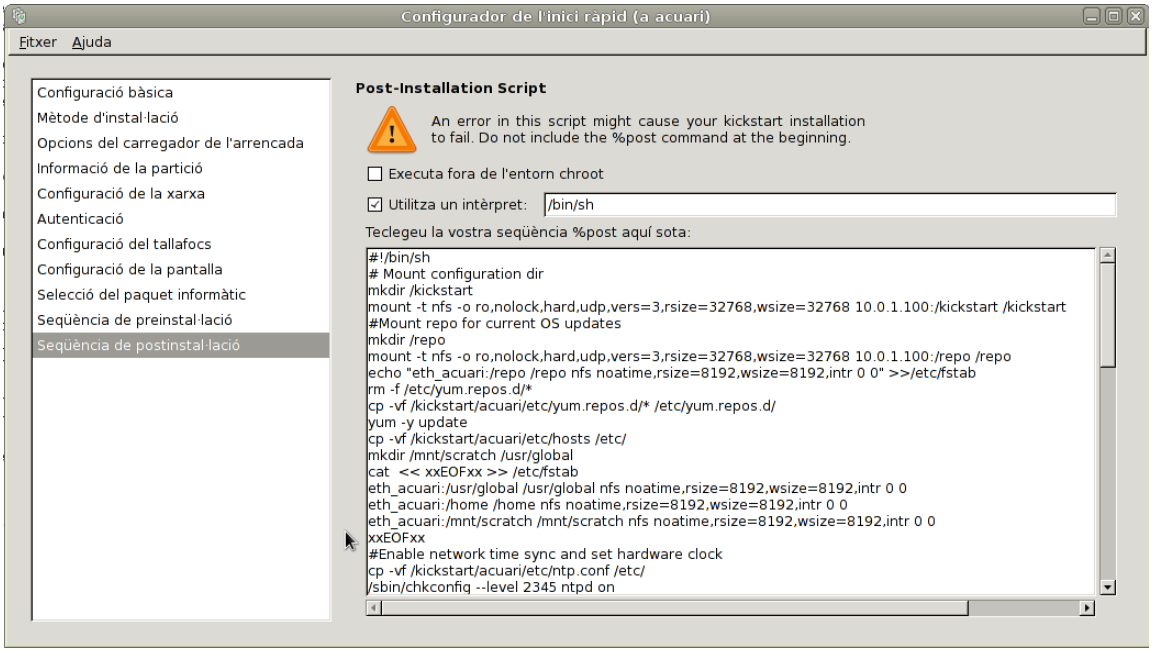


Figura 76: Configurador de Kickstart - Scripts %post

5.2.4.8 Ganglia

1. Ens baixem de la pàgina principal *http://ganglia.info* els fitxers *ganglia-monitor-core-3.2.0* i *ganglia-web-2.0-2.1.8*

Aquests fitxers tenen un .spec a dintre que haurem de modificar ja que té un bug. Només hi ha un nom de versió equivocat al .spec de ganglia-web que s'ha de corregir.

Llavors l'únic que hem de fer és:

```
rpmbuild -ta fitxer.tar.gz
```

Obtenint els respectius RPM.

2. Afegim aquests paquets PRM al repositori de cluster-packages i al node master instal·lem gmond i ganglia-web. Haurem d'instal·lar també httpd (apache2) i realitzar alguna configuració.

3. Configurem el fitxer /etc/ganglia/gmetad.conf , que és el daemon encarregat de recopilar estadístiques de tots els nodes.

4. Configurem el fitxer /etc/ganglia/gmond.conf , que és el fitxer encarregat d'enviar estadístiques a gmetad.

5. Instalem gmond a tots els nodes amb l'ordre:

   ```
   cexec yum install gmond
   ```

   Posteriorment enllaçam el fitxer de configuració /etc/ganglia/gmond.conf que haurem deixat a /globalfs/etc/ganglia/gmond.conf.

La configuració que hem realitzat és per una configuració Unicast. Això vol dir que els clients envien directament a una IP servidor i no per broadcast. Ho feim així per generar menys tràfic.

Per comprovar que funciona correctament accedirem amb un navegador extern a l'adreça d'Acuario *http://acuario.cimne.upc.edu*.

Podem trobar els fitxers de configuració als annexes A.4.6 /etc/ganglia/gmond.conf del node màster, A.4.7 /etc/ganglia/gmond.conf dels nodes esclau i A.4.8 /etc/ganglia/gmetad.conf.

5.2.5 Capa 4 – Interfícies d'administrador & usuari

5.2.5.1 C3 Tools

Executar una mateixa ordre a diversos nodes pot ser una tasca farragosa. Per això instal·larem l'eina C3 que ens permetrà disposar de la comanda “cexec”, entre altres. Aquesta comanda executa l'ordre que ve a continuació de cexec a tots els nodes definits.

Haurem d'afegir el repositori de “Ornl” i configurar C3. El paquet només s'instal·la a Acuario.

Crear /etc/yum.repos.d/ornl.repo :

```
# ORNL.repo
# PAQUET C3 TOOLS - cexec, etc.

[onrl]
name=CentOS-$releasever - ORNL_SRT
baseurl=http://bison.csm.ornl.gov/repos/rhel-5-x86_64/
gpgcheck=0
enabled=0
```

Actualitzar els fonts i instal·lar:

```
yum update
yum install c3
```

Atenció: Aquest repositori de C3 (*http://www.csm.ornl.gov/torc/C3/C3softwarepage.shtml*) ens actualitza TFTP i altres paquets. Per tant, una vegada instal·lat C3, desactivem el repositori amb enabled = 0.

Configurar /etc/c3.conf amb tots els nodes que vulguem incloure al executar la comanda:

```
[root@acuario .ssh]# cat /etc/c3.conf
cluster acuario {
      eth_acuario
      dead remove_for_0-indexing
      eth_pez001
      eth_pez002
      eth_pez003
      ...
      eth_pez015
}
```

5.2.5.2 Environment Modules

Un cop instal·lades diverses biblioteques MPI s'ha de decidir quina fer servir mitjançant variables d'entorn. En versions de RedHat anteriors a la 6.0 es feia servir la utilitat “mpi-selector”.

A partir de la versió 6 de Red Hat Enterprise Linux 6 s'utilitza el paquet “environment-modules” per tal de seleccionar quina implementació de Message Passing Interface (MPI) emprar.

La pàgina del manual aporta una bona quantitat d'informació per utilitzar aquesta eina. Un resum bàsic és el següent:

Per llistar els mòduls que hi ha disponibles al sistema executem:

```
module avail
```

Per carregar o descarregar moduls, fem servir:

```
module load <nom-de-mòdul>
module unload <nom-de-mòdul>
```

Per tal d'emular el comportament de l'antic mpi-selector, les comandes de “module load” es poden situar al nostre script d'inici de la shell (p.ex. ~/.bashrc). D'aquesta manera els mòduls es carregaran a cada accés al sistema.

Els fitxers de mòduls instal·lats es troben a:

- /usr/share/Modules/modulefiles
- /etc/modulefiles
- $HOME/privatemodules/ (només en cas de carregar el mòdul “*use.own*”)

Podem crear nous mòduls usant el llenguatge TCL i directives especials del paquet environment-modules. Consultar la pàgina del manual per més informació.

5.2.5.3 Biblioteques

Open MPI

1. Descarregar la última versió del codi des de *http://www.openmpi.org* . Hi ha dues opcions de descàrrega, la de fitxer *.tar.bz2* o la de SRPM. En aquest cas escollirem aquesta última.

   Descarregarem també l'script assistent *buildrpm.sh* tot i que no seria estrictament necessari.

   Finalment descarregarem el fitxer *.spec* que ens proporcionen en aquesta mateixa pàgina.

   Ho col·locarem tot dins un directori qualsevol, per exemple /root/src/openmpi/

2. Editarem el fitxer buildrpm.sh i afegirem a la secció "*rpmbuild_opptions=${...}*":

   ```
   --define 'install_modulefile 1'
   --define 'modules_rpm_name environment-modules'
   ```

   Modificarem també la part on s'especifica si crear l'SRPM, un RPM o múltiples RPM. Deixarem només l'opció de un sol RPM.

3. Llavors executarem:

   ```
   cd /root/src/openmpi/
   ./buildrpm.sh openmpi-1.x.x.tar.bz2
   ```

4. Si tot ha funcionat correctament els fitxers RPM s'hauràn col·locat a /usr/src/redhat o a /root/rpmbuild.

5. Finalment afegim els paquets al repositori, eliminem versions anteriors i instal·lem Open MPI:

   ```
   mv /root/rpmbuild/RPMS/* /repo/cluster-packages/x86_64/
   caterepo –update .
   yum remove openmpi && yum clean all && yum -y install openmpi
   ```

Es possible que a partir de la versió Open MPI 1.6.x hi hagi un conflicte amb libtool ja que s'intentarà sobreescriure un directori instal·lat per aquest paquet amb un executable amb el mateix nom a /usr/share/libtool.

La solució correcta en aquest cas és modificar buildrpm.sh per tal de canviar el directori d'instal·lació perquè apunti a /opt/openmpi/<versió>. D'aquesta manera a més podrem disposar de diferents versions de Open MPI centralitzades a un mateix lloc, i aprofitant l'eina environment-modules facilitar-ne el canvi entre versió i versió.

Finalment a ks.cfg haurem d'evitar la instal·lació de openmpi fins que no s'hagi acabat la configuració dels repositoris. Per tant afegirem la instal·lació als scripts de la secció %post. Si ja hi ha nodes en funcionament executarem la comanda de desinstal·lació i instal·lació amb la paraula "cexec" al principi.

Mpich2, Mvapich2, Boost, OpenMP, Blas, Lapack, Swig, Papi

Ens bastarà instal·lar-les del repositori de Scientific Linux:

```
yum update
yum install mvapich mvapich2 mpich2 blas lapack papi swig boost-openmpi boost-openmpi-python
```

En cas de voler alguna biblioteca amb una versió més moderna que la inclosa a la versió 6 de SL, haurem d'instal·lar-la manualment. Ho farem dins /opt/<pkg-name>/<version>, llavors crear un script per environment-modules i instal·lar-lo a /usr/share/Modulefiles/.

Per els nodes definirem aquests paquets a ks.cfg o executarem les comandes anterior anteposant “cexec” a la línia d'ordres.

Intel® MPI & Intel® MKL

El compilador venen en el paquet Intel® Parallel Studio XE. Les biblioteques MPI es troben en un paquet separat.

Baixarem en concret la versió de Parallel Studio XE 2011 SP1 per Linux.

1. Es descarrega la versió NON-COMERCIAL de l'Intel® Parallel Studio XE 2011 SP1 for Linux* i s'instal·len els paquets executant l'instal·lador amb *./install*. Es fa al node master. S'instal·la per defecte el software a /opt.

   Es desmarca l'Intel VTune de les opcions que se'ns donen, i es trien:

- Intel(R) Composer XE 2011 Update 6 for Linux* [None]
- Intel(R) Inspector XE 2011 Update 6 [None]
- Intel(R) VTune(TM) Amplifier XE 2011 Update 5 [All]

2. Es mou /opt/intel a /globalfs/opt/intel i es crea un enllaç simbòlic des de /opt/intel a aquest directori. Recordem que /opt dels nodes és un enllaç a /globalfs/opt.
3. Es modifica /kickstart/ks.cfg perquè crei el nou enllaç al instal·lar el nou node.
4. Ens descarreguem *l_mpi_p_4.0.3.008.tgz*, el paquet Intel MPI. El descomprimim i l'instal·lem amb l'script d'instal·lació *./install.sh* , també dins /opt/intel.
5. Creem dos nous fitxers per tal d'integrar aquest paquet amb el paquet modulefiles, aquests scripts realitzaran el que l'usuari hauria de fer manualment:

   ```
   source /opt/intel/impi/4.0.3/bin64/mpivars.sh
   source /opt/intel/bin/compilervars.sh intel64
   ```
6. Es comprova que funciona la compilació i la implementació de MPI provant amb un script. La diferència amb openmpi és que dins el .sh ara no hi ha l'ordre mpirun, sinó que el programa s'executa directament.

Adjuntem els *modulefiles* als annexos A.4.10 /etc/modulefiles/intelcc-12.1.0 i

A.4.11 /etc/modulefiles/cmake-2.8.8.

5.2.5.4 Software addicional

Ens referirem al següent software:

Emacs, Vim, Nano, Gedit, Valgrind, TCSH, GNUPlot, Python, Make.

Per instal·lar aquests paquets ho farem directament amb YUM. La única excepció serà Emacs que a la versió dels repositoris oficials hi ha un bug que fa que si es para el procés amb SIGSTOP i es surt de la sessió, el procés mort no desapareix i crea un "*memory leak*" consumint tota la memòria possible.

Per tant escarreguem emacs-23.3a.tar.gz, última versió en el moment de la instal·lació del clúster, i realitzem la instal·lació normalment.

Els altres paquets els instal·larem només al node màster:

```
yum install vim nano gedit
```

L'excepció seran els paquets tcsh, python i make que seran instal·lat a tots els nodes configurant el fitxer ks.cfg o si ja hi ha nodes funcionant amb l'eina "cexec":

```
yum install tcsh python make valgrind valgrind-openmpi
cexec yum -y install tcsh python make valgrind valgrind-openmpi
```

GiD

En aquest cas descarregarem el codi de la pàgina web de GiD: *http://gid.cimne.upc.edu* , i descomprimirem el .tar.gz dins /opt/gid/<version>. Llavors, crearem un modul que carregarà els binaris corresponents. També configurarà al següent valor la variable:

LIBGL_ALWAYS_INDIRECT=y

Per requisits de l'empresa ens poden demanar que aquest software estigui sempre disponible, per això haurem de modificar /etc/profile.d/custom.sh i /etc/profile.d/custom.csh afegint la càrrega del modul corresponent.

L'altre opció és directament posar la variable en aquest fitxer:

```
[user@acuario ~]$ cat /etc/profile.d/custom.sh
if ! echo $PATH | /bin/grep -q /opt/gid11.0 ; then
     export PATH=$PATH:/opt/gid11.0
fi
export LIBGL_ALWAYS_INDIRECT=y
[user@acuario ~]$ cat /etc/profile.d/custom.csh
setenv PATH $PATH":/opt/gid11.0"
setenv LIBGL_ALWAYS_INDIRECT y
```

Haurem de configurar el programa amb la llicència que ens proporcioni el departament de GiD.

Podem veure una captura de pantalla de GiD a la Figura 77: Software GiD.

Matlab

Seguint la mateixa dinàmica que en els programaris anteriors, descarregarem la versió de Matlab Standalone R2011b.

Instal·larem el software a /globalfs/opt/matlab/R2011b i crearem un enllaç al node màster a aquest directori dins /opt/matlab.

Crearem també un fitxer per environment-modules. Disponible a A.4.12 /etc/modulefiles/matlab-2011b.

El cas de Matlab és especial ja que requereix llicència. Per aquest motiu obtindrem la llicència i en el moment de la instal·lació indicarem on es troba el fitxer *lic_standalone.dat*.

CMake

Seguint el mateix procediment que en anteriors casos, ens baixarem la última versió de la pàgina oficial de CMake i la instal·larem a /globalfs/opt. En aquest cas no caldrà fer l'enllaç simbòlic a /opt del node màster ja que la instal·lació suporta altres directoris alternatius.

Igualment crearem un fitxer per environment-modules, el podem trobar a l'annex:

A.4.11 /etc/modulefiles/cmake-2.8.8

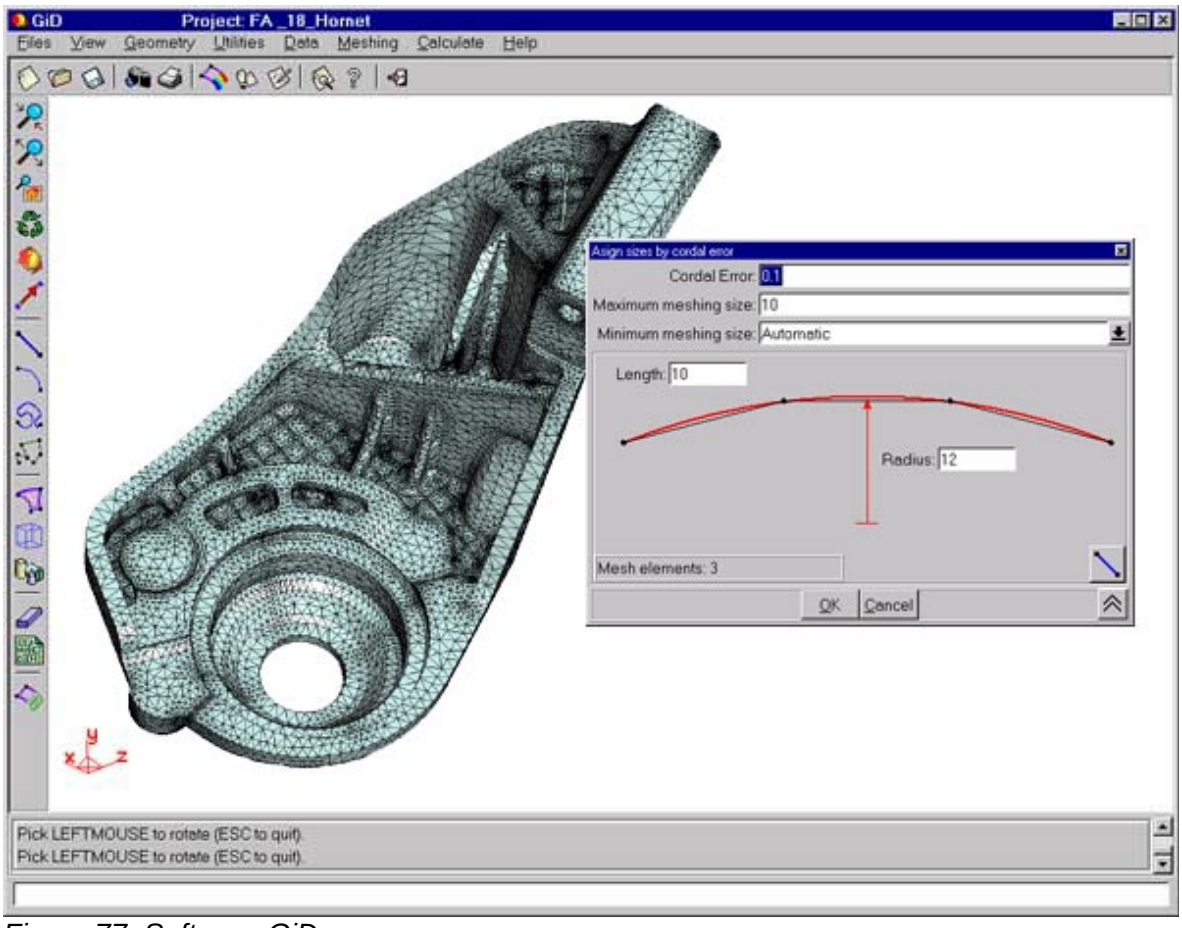


Figura 77: Software GiD

Servidor X-Window

Per tal de disposar de les biblioteques X que són dependències de paquets com Firefox o Gedit i per facilitar en algun cas puntual l'accés a un entorn gràfic al sistema físic, s'instal·larà una versió molt mínima del servidor X.

El gestor de finestres serà TWM, entorn molt mínimalista (veure Figura 78):

```
yum install xorg-x11-twm.x86_64  xorg-x11-server-Xorg.x86_64 xorg-x11-drv-ati.x86_64
```

Engegarem el servidor quan sigui necessari amb la comanda:

```
startx
```

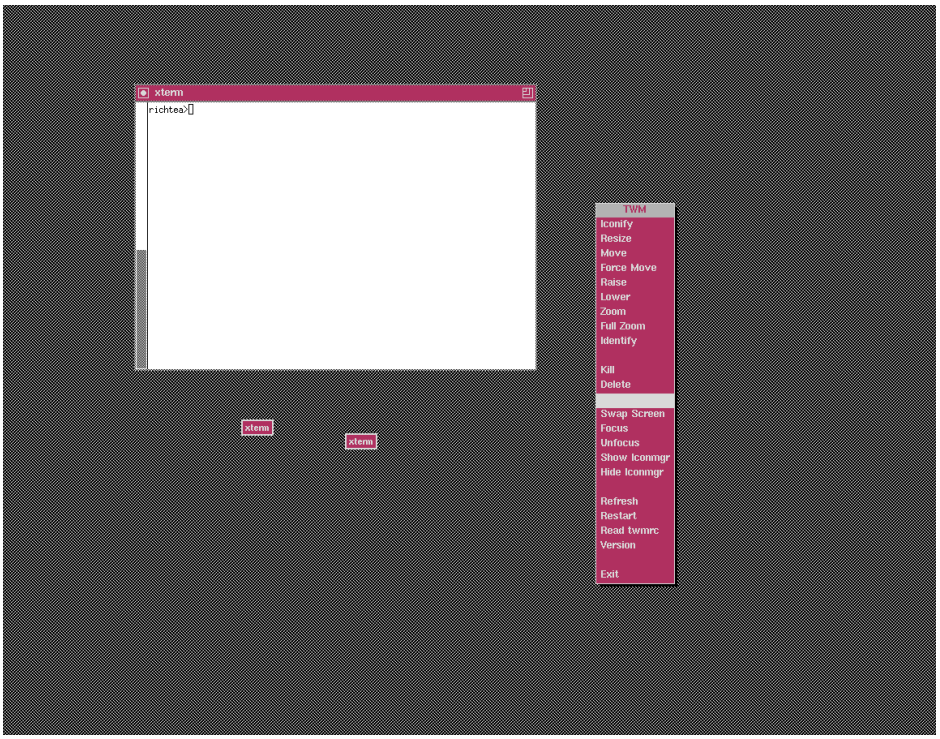
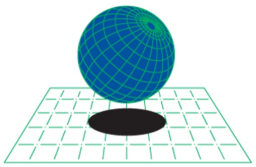


Figura 78: Aspecte del gestor de finestres TWM

Per tal de no entrar en mode gràfic quan iniciem el sistema, definirem el runlevel per defecte a 3 dins el fitxer /etc/inittab .



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 6
Desplegament

6.1 Evolució del desplegament

La planificació del desplegament de la instal·lació ha estat desenvolupada pensant principalment en interferir el mínim possible amb els treballs dels usuaris.

El marc de temps en que es realitza la implementació i posterior desplegament és de 4 mesos, des de Juny 2011 a Octubre de 2011.

En la planificació inicial es va preveure que es podria fer amb menys temps, però degut a problemes aliens es van retardar certes accions.

Per altra banda l'explicació de la implementació que hem fet en el capítol anterior ha suposat que ho fèiem a un clúster sense usuaris i no ens hem preocupat de cap aspecte relacionat.

Detallem a continuació com s'ha portat a terme tot el procés:

1. Instal·lació dels 3 nodes nous en calent, (Pez013 – Pez015) en els tres slots lliures del chassis, Juny 2011.
2. Realització de proves i instal·lacions, agafant el Pez013 com a màster amb nom “Aquari”, i els altres dos nous com a esclaus, Juliol 2011.
3. Un cop funcionant el sistema en proves es preveu capturar de forma incremental els nodes en desús, començant pels M605.

   En aquest moment succeeix un bloqueig per un usuari en el node pez002 i s'aprofita per clonar el sistema del pez013 al pez002, quedant aquest com a màster. Es decideix d'aquesta manera per proporcionar un incentiu als investigadors per tal que es passin al nou sistema, ja que els 3 nodes M610 són molt més potents que els altres, Setembre 2011.
4. Posteriorment, el 13 d'Octubre s'aconsegueixen migrar Pez010 i Pez011, nodes M605, ja disposant el nou servei de 6 nodes. La migració dura dues hores degut a alguns problemes amb DHCP.
5. El 4 de Novembre de 2011 es migra el Pez005 aprofitant que feia més d'una setmana que ningú accedia a calcular-hi.
6. El Pez003, 004, 006 i 007 es migren el 03 de Novembre de 2011, amb un temps de migració de 10 minuts. Ja només queden 4 nodes.
7. El 7 de Novembre es van migrar el Pez009 i Pez010, només queden el 008 i el màster. S'instal·len alguns programes i es resolen molts dubtes dels usuaris respecte al sistema de cues. Tot sembla funcionar correctament.
8. El 15 de Novembre de 2011 s'acaba la migració completa dels nodes.
9. Moviment dels usuaris i dades de XFire i Vega al Clúster i apagat d'aquests servidors.

Durant l'etapa s'ha hagut de muntar un cablejat alternatiu que ha involucrat la connexió d'una cabina de discs secundària de CIMNE, proporcionant volums de dades de fins a 2 TiB. Aquesta ha servit per migrar les dades de l'antic SAN, poder formatar-lo i configurar-lo.

Els comptes dels usuaris foren migrats des del primer moment mantenint les seves contrasenyes i es va configurar SSH per permetre només l'accés a certs usuaris.

Les actualitzacions de firmware de tots els components es portaren a terme durant el tall d'electricitat que sofreix el Campus Nord de la UPC el mes d'Agost.

Tots els usuaris van ser avisats de forma convenient i en el moment de les migracions ja disposaven de documentació d'usuari. També es van anar anunciant els canvis a una llista de correu i a la pàgina web *https://hpc.cimne.upc.edu* .

6.2 Validació de la instal·lació

Al acabar la instal·lació del sistema hem realitzat una sèrie de tasques bàsiques per comprovar el funcionament correcte de tots els components.

En aquest apartat donem unes indicacions de quines són aquestes tasques i també altres punts a comprovar relacionats amb les instal·lacions.

Totes les comprovacions que aquí esmentem han estat satisfactories.

6.2.1 Control d'accessos i seguretat

a) En aquest cas el comprovar que el sistema es segur es pot dur a terme amb un escaneig de ports amb *nmap*. Els únics ports que estaran oberts a la DMZ són el 9102, 22, 80 i 443. El port 80 i 443 serà accessible també des de Internet, i serà controlat també per un segon Firewall gestionat per UPCNet.

b) Realitzarem intents d'accessos per SSH amb força bruta. Equivocant-nos 3 vegades amb l'usuari root el sistema ens haurà de bloquejar. També ho farà si ens equivoquem 5 vegades amb qualsevol altre usuari.

c) Intentar canviar una contrasenya ha de donar error si no és suficientment segura:

```
[fmoll@acuario ~]$ passwd
Changing password for user fmoll.
Changing password for fmoll.
(current) UNIX password:
New password:
BAD PASSWORD: it is too simplistic/systematic
Please enter new password:
The password must have at least 6 characters.
```

d) Un usuari no pot fer ssh a cap node ni realitzar connexions cap a l'exterior del clúster que no estiguin autoritzades expressament pel firewall.

```
[fmoll@acuario ~]$ ssh pez010
fmoll@pez010's password:
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
[fmoll@acuario ~]$
```

6.2.2 Encesa, apagat i re-inici del sistema

Al reiniciar el sistema és important seguir un ordre dels equipaments tant en l'encesa com en l'apagat. Serà un punt important realitzar els següents procediments al finalitzar la instal·lació del clúster per veure si tot funciona correctament.

Encendre un sistema apagat

Suposem que el sistema es troba totalment apagat, sense ni tant sols tenir la corrent endollada.

1. Endollar la cabina de discs, el chassis i el switch.
2. Comprovar el correcte cablejat de tots els cables ethernet, veure A.3.5 Cablejat de xarxa.
3. Activar els dos interruptors posteriors del SAN i esperar uns 3 minuts a que els llums frontals estiguin en blau.
4. Engegar el chassis amb el botó del cantó inferior esquerra i esperar uns segons a que el llum del mòdul LCD estigui blau, veure 4.1.2.1 Panell de control frontal.
5. Engegar el node Acuario i comprovar que s'han muntat els directoris /mnt/cluster-data i /home corresponents als volums de la cabina de discs. Comprovar també que les interfícies de xarxa estan actives.
6. Engegar els nodes pez001 a pez015.
7. Comprovar que com a usuari root podem fer ssh als nodes, i que apareixen a SLURM i a Ganglia tots els nodes.
8. Comprovar que no hi hagi cap reserva de nodes per manteniment a SLURM. Eliminar-la des de “sview” o “sacctmgr” si cal.

Apagar un sistema encès

Suposem que el sistema es troba encès i ja s'ha avisat a tots els usuaris i planificat l'apagat.

1. Des de Acuario executar l'ordre “cexc halt” que apagarà tots els nodes.
2. Executar llavors l'ordre “halt” i esperar que el node Acuario s'apagui.
3. Desactivar els interruptors del SAN.
4. Apagar el chassis amb el botó del cantó inferior esquerra.
5. Desendollar si cal el chassis, el switch i la cabina de discs.

Re-inici o apagat de certs nodes

Suposem que el sistema es troba encès i ja s'ha avisat a tots els usuaris i planificat l'apagat.

1. En el cas de requerir el re-inici o apagat d'alguns nodes individuals, s'haurà d'executar per cada node que vulguem reiniciar:

   ```
   ssh pez0[01-15] <reboot,halt>
   ```

Si el procediment no funciona perquè l'accés al node és impossible, executar un navegador web des del node màster i entrar a la consola d'administració del chassis (CMC), IP 172.26.0.200 tal com especifiquem a 5.2.3.7 Dell CMC & iDrac. Donar l'ordre de *shutdown* des d'aquesta consola.

Re-inici del sistema

Suposem que el sistema es troba encès i ja s'ha avisat a tots els usuaris i planificat el re-inici.

1. Realitzar l'apagat de tots els nodes excepte Acuario executant la següent ordre:

   ```
   cexec halt
   ```

2. Reiniciar o apagar Acuario en funció de si volem reiniciar també el SAN o el Switch, amb l'ordre halt o reboot.
3. Procedir a l'engegat del sistema com s'especifica a Encendre un sistema apagat.

6.2.3 Restauració de nodes

Realitzarem un re-inici d'un node que no s'estigui utilitzant i farem que arranqui per PXE. Realitzarem el desplegament de la imatge per TFTP.

El funcionament ha estat satisfactori excepte en determinats moments en que el DHCP tarda més de 45 segons en donar una IP i NetworkManager d'Anaconda mor per un bug. Es comenta més endavant a l'apartat de problemes.

El procés, quan funciona correctament (normalment a la segona vegada si ha fallat prèviament), instal·la el sistema complet i el posa en funcionament amb menys de 10 minuts.

6.2.4 Actualitzacions

La política d'actualitzacions ha de ser molt estricte i només portar-se a terme quan realment sigui necessari. Se seguirà el següent procediment:

1. En el moment de trobar una actualització d'algun component, comprovar si aquesta afecta a la nostra instal·lació i configuració particular.
2. Si afecta i és critica:

   Planificar el moment de l'actuació i si interromp els treballs dels usuaris avisar-los amb suficient antelació i quedar d'acord amb que fer.

   Si no és crítica:

   Esperar al tall d'electricitat del mes d'Agost per portar a terme l'actualització.
3. Abans d'actualitzar tots els nodes i el node màster és important provar les actualitzacions en un sol node. Per exemple un canvi de versió del kernel pot dur problemes amb algun driver, i realment passa.

Per altra banda descrivim a continuació una prova d'actualització de tota la pila de software del clúster realitzada satisfactòriament. Aquesta operació és molt arriscada tot i que certament possible.

És obligatori consultar prèviament la pàgina de Scientific Linux que parla sobre l'actualització del sistema:

http://www.scientificlinux.org/documentation/howto/upgrade.6x

Procediment:

1. Realitzar còpies de seguretat d'Acuario.
2. Assegurar-nos que hi ha suficient espai de disc, en particular al directori /var/cache que és on es descarreguen els RPMS.
3. Modificar els repositoris /etc/yum.repos.d/sl* desactivant el repositori local, i activant els repositoris d'Internet.
4. **yum update**

   Actualitzarà el sistema amb els darrers paquets de la 6.X, cosa recomanable. D'aquesta manera es farà l'últim pas més ràpid.
5. **yum clean all**

   Neteja totes les capçaleres i paquets de la antiga configuració de yum. Si no ho fas és possible que yum en passos posteriors et digui que no hi ha res per fer.
6. **yum --releasever=6.Y update sl-release**

   Nota: 6.Y pot ser substituït amb qualsevol versió que vulguis anar, per exemple la 6.2 Això instal lara la última versió del paquet sl-release al sistema. sl-release et diu quina versió de SL tens instal lada.
7. **Updating from 6.0 only yum install yum-conf-sl-other**

   A SL 6.0, 'security' i 'fastbugs' eren tots dos al fitxer sl-updates.repo, proporcionat pel paquet rpm sl-release. A SL 6.1, 'security' està ara dins el fitxer sl.repo, proporcionat pel paquet rpm sl-release, i 'fastbugs' dins el fitxer sl-other.repo, proporcionat pel paquet rpm yum-conf-sl-other.
8. **yum update**

   En aquest pas és on s'actualitza tot el sistema.
9. Comprova les configuracions de grub. Encara que yum normalment ho fa tot bé quan actualitza els kernels, sempre es bona idea revisar el fitxer de grub per si un cas.
10. optional **yum clean all**

    Això neteja tots els rpms que t'has descarregat no malgastant l'espai de disc.
11. Crea un nou repositori local amb la nova versió del la distribució. Veure l'apartat 5.2.4.2 Repositori de software YUM. Suposem que el creem dins /repo/scientific/6.Y/.
12. Actualitzem /mnt/cluster-data/sbin/update-repositoris.sh canviant 6.X per 6.Y.
13. Descarregar l'Install DVD de la 6.Y dins /repo/scientific/6.Y/x86_64/iso/.
14. Modificar les imatges del kernel de TFTPBoot per les del nou DVD tal com s'explica a 5.2.4.7 Kickstart + PXE (/tftpboot/images/scientific/6.Y/x86_64/). Editar també les entrades del menú canviant 6.X per 6.Y (/tftpboot/pxelinux.cfg/default).
15. Editar /kickstart/ks.cfg i canviar tota referència al repositori 6.X, posant-hi 6.Y.
16. **/sbin/reboot**

    Reinicia el sistema a la versió de Scientific Linux actualitzada.
17. Modificar els repositoris de YUM perquè apuntin de nou al repositori local.
18. Actualitzar el repositori local amb l'script /mnt/cluster-data/sbin/update-repositoris.sh

19. Per actualitzar els nodes és més ràpid i net fer una instal·lació nova, per tant els engegem iniciant el boot per defecte a PXE Boot i carregant el nou sistema operatiu.

Hi ha diverses formes d'actualitzar el sistema, però aquesta és la més eficaç i ràpida. Tot i així aquesta forma no conserva la possibilitat de mantenir el repositori de la 6.X i defensa un clúster unificat a una mateixa versió. És possible amb una mica més de feina fer instal·lacions personalitzades.

Atenció, pot haver-hi problemes:

Per exemple, nous noms de les interfícies de la versió 6.0 a la 6.1:

A partir de la versió 6.1 de RedHat Linux (i per tant de SL 6.1), els servidors Dell implementen un nou esquema de nomenclatura de la xarxa. S'utilitzen ara noms proporcionats per la BIOS per definir les interfícies.

Com que en el nostre clúster estan ben definides les interfícies, i com que pensant en el futur es possible afegir nodes externs que no siguin Dell, hem d'evitar utilitzar la nova nomenclatura.

La forma de fer-ho és posant com a opció al kernel “biosdevname=0”. D'aquesta manera ja no s'utilitza la nova nomenclatura.

Això fa que el menús de PXE Boot s'hagin de modificar.

http://linux.dell.com/files/whitepapers/consistent_network_device_naming_in_linux.pdf

http://en.community.dell.com/dell-blogs/enterprise/b/tech-center/archive/2011/05/26/meaningful-names-for-network-devices-in-rhel-6-sp1-on-dell-systems.aspx

https://fedoraproject.org/wiki/Features/ConsistentNetworkDeviceNaming

6.2.5 Compilació de programes

Comprovem que els compiladors instal·lats són capaços de compilar programes sèrie, OpenMP i MPI. Suposem el nom del codi font “hello-world-*.c”:

Compilador GNU GCC 4.4.5

```
[fmoll@acuario gcc]$ gcc -o hello-world.serial hello-world-serial.c
[fmoll@acuario gcc]$ gcc -o hello-world.omp hello-world-omp.c -fopenmp
[fmoll@acuario gcc]$ module load openmpi-x86_64
[fmoll@acuario gcc]$ mpicc -o hello-world.mpi hello-world-mpi.c
```

Compilador Intel 12.1.0

```
[fmoll@acuario intl]$ module load intelcc-12.1.0
[fmoll@acuario intl]$ icc -o hello-world.serial hello-world-serial.c
[fmoll@acuario intl]$ icc -o hello-world.omp hello-world-omp.c -openmp
[fmoll@acuario intl]$ module load impi-4.0.3
[fmoll@acuario intl]$ mpicc -o hello-world.omp hello-world-mpi.c
```

Podem trobar exemples de codi OpenMP i Open MPI a l'apartat 4.3 Models de programació paral·lela.

6.2.6 Execució de processos

Un cop compilats els anteriors programes, podem executar-los amb srun si són sèrie o OpenMP, o amb sbatch si també volem MPI. La forma recomanada és sempre utilitzar scripts i llançar-los amb sbatch.

Treballs en sèrie

En aquest exemple demanem executar una sola tasca (--ntasks), i que cadascuna sigui enviada a un nucli (--ntasks-per-core), amb una memòria per nucli (--mem-per-cpu) de 3 GiB i a la partició Short, per un temps d'execució de màxim 5 minuts (--time).

El nom del treball serà JobName, i la sortida estàndard es gravarà a JobName-output-job_%j.out, mentre que l'error al mateix nom acabat en “.err”.

```
[fmoll@acuario intl]$ cat run.sh
#!/bin/bash
#SBATCH --job-name=JobName
#SBATCH --output=JobName-output-job_%j.out
#SBATCH --error=JobName-output-job_%j.err
#SBATCH --ntasks-per-core=1
#SBATCH --ntasks=1
#SBATCH --partition=Short

##Optional - Required memory in MB per core. Defaults is 3GB per core.
#SBATCH –mem-per-cpu=3072

##Optional - Estimated execution time
#SBATCH --time="5"

########### Further details -> man sbatch ##########
cd /home/user/binaries/
./executable
```

Per enviar el treball a SLURM executem:

```
[fmoll@acuario test]$ sbatch run.sh
srun: jobid 2314 submitted
```

Treballs OpenMP

En aquest exemple demanarem espai per 8 fils d'execució, un a cada nucli. Per això utilitzem l'opció *--ntasks-per-node*. Llançarem el treball a la partició XeonE5645 i en total no volem ocupar més de 3GiB.

```
#!/bin/bash
#SBATCH --job-name=JobName
#SBATCH --output=JobName-output-job_%j.out
#SBATCH --error=JobName-output-job_%j.err
#SBATCH --partition=XeonE5645
#SBATCH --ntasks-per-node=8
#SBATCH --mem=3072

########### Further details -> man sbatch ##########
export OMP_NUM_THREADS=8
cd /home/user/openmp-binary/
./binary
```

Per enviar el treball a SLURM executem:

```
[fmoll@acuario test]$ sbatch run.sh
srun: jobid 2315 submitted
```

Treballs Open MPI

En aquest exemple demanarem executar 36 tasques, un a cada nucli, no importa en quins nodes. Llançarem el treball a la partició XeonE5645. Per aquest tipus de tasques només podem fer servir salloc o sbatch. El paràmetre --ntasks serà enviat a “mpirun” com a “mpirun -np <ntasks>”.

Primer de tot carregarem el mòdul de Open MPI:

```
[fmoll@acuario test]$ module load openmpi-x86_64
```

Script run.sh:

```
#!/bin/bash
#SBATCH --job-name=JobName
#SBATCH --output=JobName-output-job_%j.out
#SBATCH --error=JobName-output-job_%j.err
#SBATCH --partition=XeonE5645
#SBATCH --ntasks-per-core=1
#SBATCH --ntasks=36

########### Further details -> man sbatch ##########
cd /home/user/mpibinary/
mpirun myMPIexecutable
```

Treballs Intel MPI

Igual que en el cas anterior demanarem executar 36 tasques, un a cada nucli, no importa en quins nodes. Llançarem el treball a la partició XeonE5645. Per aquest tipus de tasques podem fer servir salloc, sbatch o srun.

Primer de tot carregarem el mòdul de Intel MPI:

```
[fmoll@acuario test]$ module load impi-4.0.3 intelcc-12.1.0
```

Script run.sh:

```
#!/bin/bash
#SBATCH --job-name=JobName
#SBATCH --output=JobName-output-job_%j.out
#SBATCH --error=JobName-output-job_%j.err
#SBATCH --partition=XeonE5645
#SBATCH --ntasks-per-core=1
#SBATCH --ntasks=36

########### Further details -> man sbatch ##########
cd /home/user/mpibinary/
srun myMPIexecutable
```

6.2.7 Monitoreig de processos

Comprovarem que els processos enviats a SLURM apareguin a les cues i es puguin modificar només per l'usuari propietari i root. Provarem les eines sview, scontrol, smap, sacct.

També provarem la concurrència entre processos observant quins nuclis agafa cada treball d'usuari i mirant que dos treballs no agafin el mateix recurs. Per fer-ho podrem utilitzar tant les eines de SLURM (sacct, sinfo...) com entrar per SSH al node en qüestió i comprovar amb “top” quins nuclis ha agafat cada procés.

També comprovarem que el monitor de Gànglia funciona correctament accedint a la pàgina web *http://acuario.cimne.upc.edu* .

6.2.8 Rendiment Preliminar

En aquest apartat hem realitzat una sèrie de proves per tal de veure quin rendiment global obteníem en els punts més crítics del sistema. Aquests són bàsicament l'accés a disc, tant local com per NFS, i la comunicació per la xarxa Infiniband.

6.2.8.1 Ample de banda de disc

Hem realitzat unes proves bàsiques per veure que el sistema funciona normalment. El procediment ha consistit en escriure amb l'eina dd 10GB de dades en un fitxer a disc local, un altre al disc local muntat per iSCSI i finalment un altre des de node remot que té muntada la unitat NFS.

El resultat ha estat coherent en els dos primers casos, mentre que en el tercer s'ha mostrat una baixada molt notable del rendiment, de fins a un 30%. Suposem que aquest problema és degut a diversos factors, entre ells els següents:

- En el moment de l'execució del test ho vam fer amb usuaris treballant al sistema, essent la càrrega de la xarxa elevada i compartida entre un total de 44 processos que usaven NFS.

  A la Figura 79 i Figura 80 podem veure com en el moment de l'execució del test, aproximadament a les 20h, en el Pez002 comença a realitzar-se la prova. En aquest mateix moment, en tots els altres nodes comença a haver-hi una pujada de la càrrega de la xarxa, factor que ha influït notablement en el rendiment.

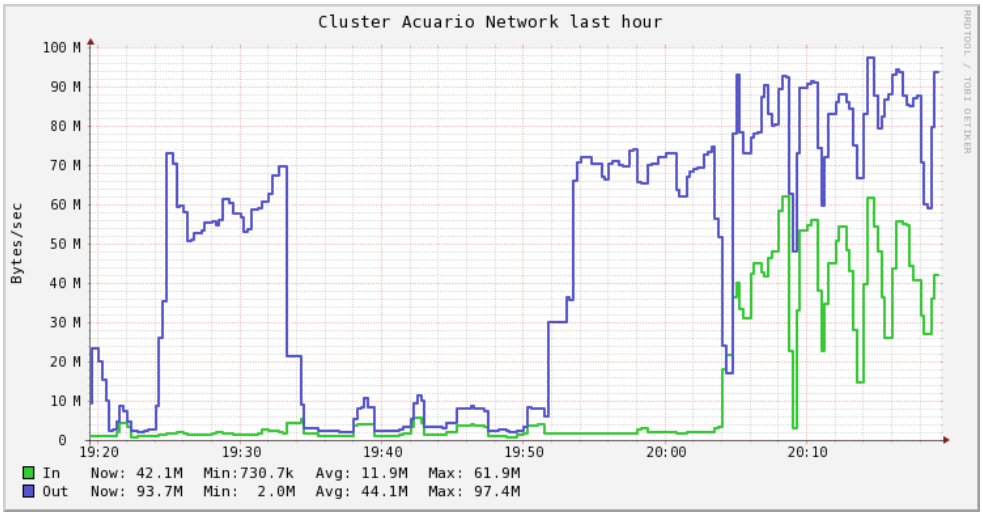


Figura 79: Càrrega de xarxa global del Clúster en el moment de fer el test NFS

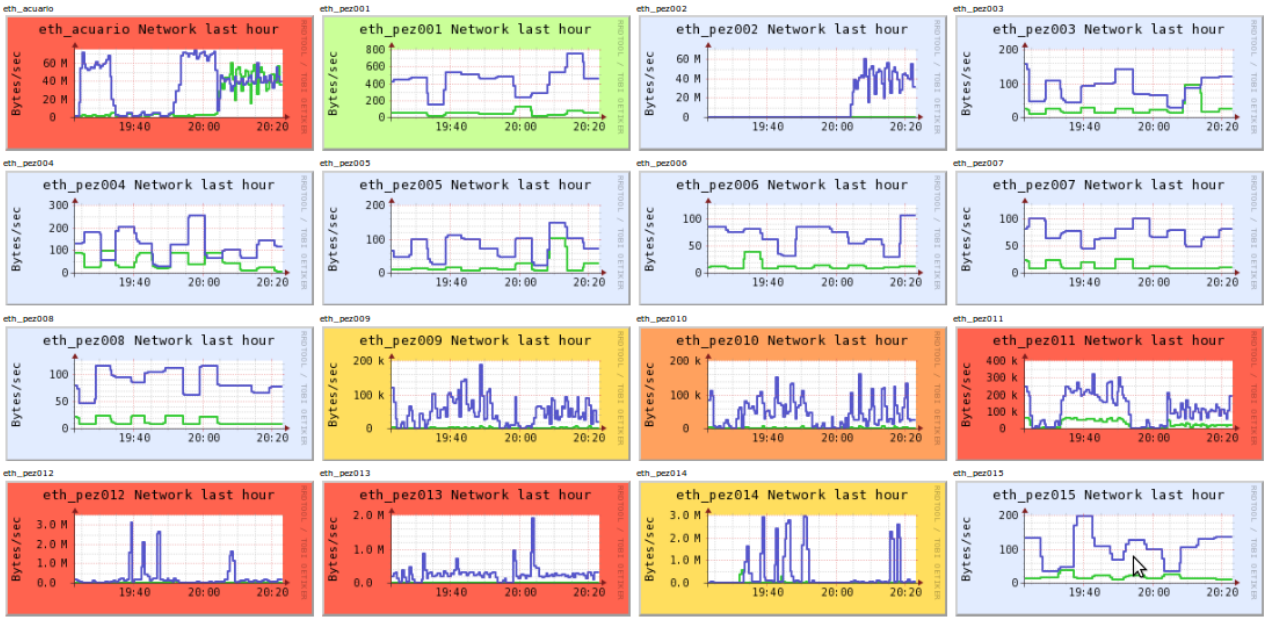


Figura 80: Càrrega de xarxa en el moment de realitzar el test NFS

- Un altre factor que pot haver influït en el resultat és algun paràmetre no optimitzat de NFS. En el futur investigarem quines opcions de millora tenim. Ens pot donar una pista el fet que el millor rendiment s'obtingui amb blocs de 32KB.

El resultat de les execucions es mostra a la Figura 81 i Taula 11:

Mida bloc (KB)	***Disc local (/tmp)***	***Disc ISCSI (Acuario)***	***Disc NFS (Pez002)***
128KB	*162 MB/s*	*122 MB/s*	*36,5 MB/s*
64KB	*163 MB/s*	*122 MB/s*	*38,7 MB/s*
32KB	*165 MB/s*	*137 MB/s*	*39,6 MB/s*
4KB	*75 MB/s*	*68,2 MB/s*	*39,4 MB/s*

Taula 11: Resultats escriptura a disc, iSCSI i NFS

No obstant això i a la vista dels resultats, podem dir que la cabina de discs es comporta correctament ja que ens dona l'ample de banda propi d'una connexió d'1Gb, aproximadament 125MB/s. Per altra banda la gran utilització del clúster en els moments de fer les proves ha fet impossible detectar si es traca d'un problema de NFS o es un a baixada del rendiment temporal.

Una altra conclusió que podrem treure és que la concurrència d'usuaris afecta molt negativament a l'ample de banda i que en cas d'haver-hi investigadors que es queixin hem de tenir en compte aquest factor i recomanar-lis el calcular en disc local.

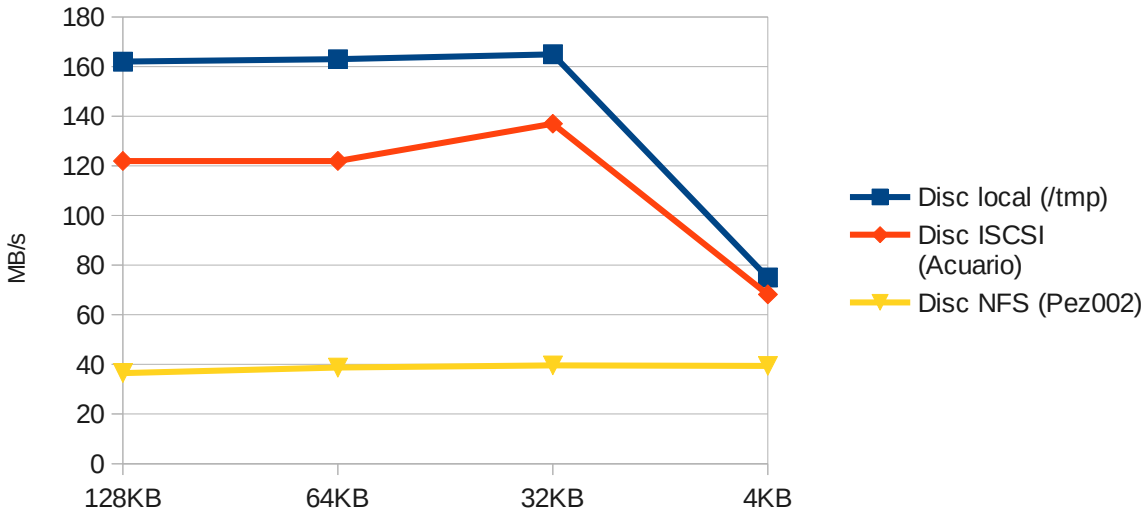


Figura 81: Comparació del rendiment entre muntatges de sistemes de fitxers

6.2.8.2 Ample de banda Infiniband

En aquesta prova hem volgut comprovar que la velocitat dels enllaços Infiniband és de 20Gbps entre HCAs DDR, i de 40Gbps entre els QDR. Aquesta velocitat ens ha de permetre arribar a 2500MB/s i 4000MB/s teòrics respectivament.

Primer de tot mirem si hi ha connectivitat entre dos nodes, posem en aquest exemple Pez002 i Pez003. Prèviament hem hagut d'augmentar la freqüència de CPU de tots els nuclis modificant els dispositius de cpufreq “scaling_governor” amb el valor “performance”, [57]:

Executem ibv_rc_pingpong al Pez002:

```
[root@pez002 ~]# ibv_rc_pingpong
  local address:  LID 0x000b, QPN 0x16004a, PSN 0xc31980
```

Executem posteriorment ibv_rc_pingpong al Pez003 amb el nom de host del servidor remot:

```
[root@pez003 ~]# ibv_rc_pingpong pez002
  local address:  LID 0x0002, QPN 0x56004a, PSN 0x6f9349
  remote address: LID 0x000b, QPN 0x16004a, PSN 0xc31980
8192000 bytes in 0.02 seconds = 4181.99 Mbit/sec
1000 iters in 0.02 seconds = 15.67 usec/iteer
```

Si la connexió és satisfactòria es mostrarà la informació del procés en ambdós extrems.

Pez002:

```
[root@pez002 ~]# ibv_rc_pingpong
  local address:  LID 0x000b, QPN 0x16004a, PSN 0xc31980
  remote address: LID 0x0002, QPN 0x56004a, PSN 0x6f9349
8192000 bytes in 0.02 seconds = 4147.32 Mbit/sec
1000 iters in 0.02 seconds = 15.80 usec/iter
```

Per comprovar l'ample de banda només ens cal executar la comanda ib_read_bw o ib_write_bw de la mateixa forma que ho hem fet anteriorment. En aquesta ocasió ho farem entre el node Pez002 i el Pez003, entre Pez002 i el Pez014, i entre Pez013 i Pez014.

La opció -b indica que es realitzin tests bi-direccionals.

Pez002:

```
[root@pez002 ~]# ib_read_bw -b
---------------------------------------------------------------------
                    RDMA_Read Bidirectional BW Test
Connection type : RC
  local address:  LID 0x0b, QPN 0x1a004a, PSN 0x209f2a RKey 0xe6002186
VAddr 0x007f0a42c93000
    remote address: LID 0x02, QPN 0x5c004a, PSN 0x9492c2, RKey
0x1a002161 VAddr 0x007fc6cf5c0000
Mtu : 2048
---------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        1000            2527.17               2527.15
---------------------------------------------------------------------
```

Pez003:

```
[root@pez003 ~]# ib_read_bw -b pez002
---------------------------------------------------------------------
                    RDMA_Read Bidirectional BW Test
Connection type : RC
  local address:  LID 0x02, QPN 0x5c004a, PSN 0x9492c2 RKey 0x1a002161
VAddr 0x007fc6cf5c0000
    remote address: LID 0x0b, QPN 0x1a004a, PSN 0x209f2a, RKey
0xe6002186 VAddr 0x007f0a42c93000
Mtu : 2048
---------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        1000            2312.74               2064.73
---------------------------------------------------------------------
```

Ara provem entre el node Pez013 i Pez014:

Pez013:

```
[root@pez013 ~]# ib_read_bw -b pez014
---------------------------------------------------------------------
                    RDMA_Read Bidirectional BW Test
Connection type : RC
   local address:  LID 0x10, QPN 0x4004a, PSN 0xd26024 RKey 0x4002100
VAddr 0x007f17f9e69000
  remote address: LID 0x0f, QPN 0x4004a, PSN 0x7d94a2, RKey 0x4002100
VAddr 0x007fcc23f15000
Mtu : 2048
---------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        1000            3713.74               3713.73
---------------------------------------------------------------------
```

Pez014:

```
[root@pez014 ~]# ib_read_bw -b
---------------------------------------------------------------------
                    RDMA_Read Bidirectional BW Test
Connection type : RC
  local address:  LID 0x0f, QPN 0x4004a, PSN 0x7d94a2 RKey 0x4002100
VAddr 0x007fcc23f15000
  remote address: LID 0x10, QPN 0x4004a, PSN 0xd26024, RKey 0x4002100
VAddr 0x007f17f9e69000
Mtu : 2048
---------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        1000            3717.41               3717.40
---------------------------------------------------------------------
```

Finalment entre Pez002 i Pez014:

Pez014:

```
[root@pez014 ~]# ib_read_bw -b
---------------------------------------------------------------------
                    RDMA_Read Bidirectional BW Test
Connection type : RC
  local address:  LID 0x0f, QPN 0x76004a, PSN 0xff1f13 RKey 0xca002112
VAddr 0x007f752c54a000
    remote address:  LID 0x0b, QPN 0x1c004a, PSN 0x9bc744, RKey
0xe8002186 VAddr 0x007fd12aaf0000
Mtu : 2048
---------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        1000            2079.17               2051.03
---------------------------------------------------------------------
```

Pez002:

```
[root@pez002 ~]# ib_read_bw -b pez014
---------------------------------------------------------------------
                    RDMA_Read Bidirectional BW Test
Connection type : RC
  local address:  LID 0x0b, QPN 0x1c004a, PSN 0x9bc744 RKey 0xe8002186
VAddr 0x007fd12aaf0000
    remote address:  LID 0x0f, QPN 0x76004a, PSN 0xff1f13, RKey
0xca002112 VAddr 0x007f752c54a000
Mtu : 2048
---------------------------------------------------------------------
 #bytes #iterations    BW peak[MB/sec]    BW average[MB/sec]
  65536        1000            2617.77               2526.13
---------------------------------------------------------------------
```

Ens hem de fixar en el valor de la mitja d'ample de banda obtinguda. Veiem que en el primer cas, entre dos nodes amb HCA DDR disposem d'una mitja entre 2000 i 2500MB/sec. En el segon cas amb HCA QDR obtenim el valor de l'ordre de 3700MB/sec.

En el tercer cas, quan mesclem nodes amb HCA QDR i DDR, obtenim la velocitat DDR de l'ordre de 2500MB/s.

En tres casos obtenim un rendiment correcte i molt semblant al marge teòric esperatm, per tant assumim que la instal·lació ha estat realitzada correctament.

6.2.9 Planificador de treballs

Per comprovar que el planificador de treballs funciona de forma correcta, llançarem diversos treballs amb diversos usuaris al mateix temps i comprovarem que són inserits en ordre a la cua.

També provarem el mecanisme de backfill enviant un treball gran, un de mitjà i un altre de petit. Haurem de comprovar que el treball més petit passa davant el mitjà.

Provarem també de modificar les prioritats de certs usuaris o particions i veurem com els treballs s'ordenen en funció de la prioritat.

Finalment consultant les estadístiques dels treballs veurem com el fairshare actua correctament:

```
[root@acuario modulefiles]#  sshare -a
             Account       User Raw Shares Norm Shares   Raw Usage Effectv Usage  FairShare
-------------------- ---------- ---------- ----------- ----------- ------------- ----------
root                                          1.000000    53307274      1.000000   0.500000
 root                       root          1    0.333333         898      0.000017   0.999965
 default                                  1    0.333333    53035165      0.994895   0.126334
  default                  alara          1    0.006667           0      0.019898   0.126334
  default                anaddia          1    0.006667        2767      0.019949   0.125667
  default                 aseret          1    0.006667           1      0.019898   0.126334
  default                cafiero          1    0.006667          32      0.019898   0.126326
  default                camddos          1    0.006667           0      0.019898   0.126334
  default               cdabalos          1    0.006667     1938749      0.055540   0.003105
  default                  fmoll          1    0.006667         589      0.019909   0.126192
```

6.3 Problemes

Durant la instal·lació i el desplegament del servei han aparegut alguns problemes que es comenten a continuació.

6.3.1 Desplegament amb PXE + Kickstart

Al realitzar el desplegament d'un node amb PXE + Kickstart, tal com tenim configurat a ks.cfg s'inicia la instal·lació de Scientific Linux 6.1. Un dels primers passos que realitza l'instal·lador de RHEL Anaconda és el de llançar el NetworkManager per obtenir adreçes IP de la xarxa.

NetworkManager prova d'establir la configuració d'eth0 segons ks.cfg, però al esperar la IP que serà donada per el servidor DHCP d'Acuario esdevé el “timeout” de 45 segons per defecte i la instal·lació s'avorta.

El timeout ha de ser modificable des del paràmetre d'inici del kernel que podem especificar a /tftpboot/pxelinux.cfg/default, no obstant això, al fer-ho NetworkManager l'ignora i avorta igualment.

Aquest problema és un bug documentat a RHEL del qual no havien donat cap solució fins a 20-06-2012. La causa freqüent és que s'utilitzen cert tipus de switchs (com el Dell PowerConnect) que retrassen alguns tipus de paquets.

Hem provat de canviar les opcions de ks.cfg de --bootproto=dhcp per --bootproto=bootp sense èxit.

- El bug és el #663820: *https://bugzilla.redhat.com/show_bug.cgi?id=663820*
- RedHat ha creat una solució que serà aplicada properament RHBA-2012-0832.

  http://rhn.redhat.com/errata/RHBA-2012-0832.html

Com a forma temporal que funciona, és reiniciar algunes vegades el node fins que en una d'elles aconsegueix agafar IP i continuar endavant.

Com a forma definitiva és instal·lar el paquet per RHEL 6.3 que soluciona el bug #663820:

https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/6.3_Technical_Notes/NetworkManager.html#RHBA-2012-0832

6.3.2 Emacs memory lake

La versió de Emacs del repositori de paquets oficial de Scientific Linux 6.1 conté un bug.

Quan s'envia un SIGSTOP al programa i llavors es surt de la sessió, Emacs no mor i comença a consumir memòria RAM fins al limit.

Aquest problema ens va portar alguns mal de caps perquè es va donar quan encara no teniem implementats els límits per usuari en el node màster, fet que feia que morissin alguns processos com els de SLURM i les cues deixessin de funcionar.

Vam realitzar investigacions fins a trobar la causa del problema veient com el sistema matava alguns processos segons informació de /var/log/messages:

```
Out of memory: kill process 4757 (bash) score 517695 or a child
...
Out of memory: kill process 18317 (slurmctld) score 404598 or a child
...
Out of memory: kill process 3337 (dsm_om_connsvcd) score 318477 or a child
...
Out of memory: kill process 4471 (emacs) score 286922 or a child
```

La solució va passar per baixar la última versió de Emacs de la pàgina web oficial i instal·lar-la manualment al node màster.

6.3.3 Límits dels recursos al node màster

El node màster estava pensat per ser un senzill node d'accés al servei de càlcul. No obstant això, ens han demanat en diverses ocasions que el poguessin fer servir per visualitzar els resultats de les execucions ja que aquests ocupaven molt d'espai i fer la transferència per xarxa habitualment no era possible.

Al permetre aquesta opció hem hagut d'investigar quines mesures havíem de prendre per no permetre que els usuaris sobrepassessin els límits del sistema. Les opcions disponibles són les d'implementar uns límits plans, amb un màxim de recursos per cada usuari.

Després de l'experiència obtinguda creiem que seria bo disposar d'un sistema dinàmic d'assignació de recursos al node màster, ja que si per exemple limitem com hem fet, a X GiB de memòria / usuari, hem de preveure quin serà l'ús màxim de cada usuari, i podem cometre errors.

És a dir, si tenim 32 GiB al sistema, i 32 usuaris, tocarà limitar equitativament a 1GiB per usuari, però és possible que dels 32 usuaris només n'hi hagi 5 que faixin servir realment 1 GiB i per tant l'espai sobrant d'aquests hauria de poder ser assignat a altres usuaris.

Aquest és un problema que encara no hem resolt i que queda com a tasca futura.

6.3.4 Integració amb LDAP

Al principi es volien integrar les comptes amb l'actual sistema LDAP de CIMNE. El problema que vam trobar i que ens ho va impedir va ser que tots els usuaris pertanyien al mateix grup degut a causes alienes a nosaltres. Aquesta configuració hagués causat una complicació en quant a gestió d'usuaris i grups i a més com que des de Sistemes se'ns va dir que properament es modificaria el servidor LDAP vam preferir deixar un sistema alternatiu que esdevingué NIS.

5.2.4.5 Sincronització d'usuaris, grups i hosts.

6.3.5 Integració amb el sistema Icinga (Nagios)

La planificació inicial comptava amb poder integrar la monitorització del clúster amb el nou sistema Icinga que s'implementava a CIMNE. A la data de finalització d'aquest projecte, el nou servei encara no estava actiu i va fer impossible complir l'objectiu.

Com a mètode alternatiu de monitorització hem instal·lat les eines de Dell OMSA i també hem configurat l'accés a l'iDrac. Finalment hem proporcionat accés a l'estat i configuració del SAN i del switch.

Amb aquestes eines i també amb Ganglia podem veure l'estat complet dels servidors, el chassis, el SAN, el switch i en definitiva tots els components que tenen a veure amb el servei instal·lat.

En ocasions futures es preveu integrar el sistema amb el nou Icinga.

6.3.6 Queixes dels usuaris

En certes ocasions havíem planificat fer la migració d'alguns nodes que no han estat possibles en les dates assenyalades. Això és degut a que els usuaris estaven executant processos i tot i els avisos donats ens han insistit que no aturessim els nodes perquè volien calcular més.

Hi ha hagut escenes de tot tipus i en general hem trobat alguns usuaris no col·laboratius en aquest aspecte.

Per altra banda també se'ns han queixat alguns altres de que no es pogués fer SSH als nodes i que se'ls canviés la forma de treballar. No entenien que era per un millor aprofitament de tots els recursos i una millora real en la productivitat.

Altres problemes que també hem pogut tenir és que a l'hora d'investigar quines eren les millors solucions pels usuaris en quant a la gestió dels treballs aquests no s'han posat d'acord i hi han hagut discussions. Al final s'ha optat per prendre el control i implementar la política que més s'ajustava a tots els esquemes, la de backfill. Problemes similars hem tingut en la definició de les particions de nodes, on alguns defensaven particions petites, altres grans, i altres cap partició o límit de temps.

En definitiva però s'ha arribat a un consens i sembla a ser que en general hom ha quedat content una vegada provat el nou sistema i les eines gràfiques que aquest inclou.

6.3.7 Retard en l'escriptura NFS

Alguns usuaris se'ns han queixat de que no podien veure els resultats en temps real dels seus processos. Aquests usuaris accedien a fitxers de disc que eren escrits mentre el seu programa s'executava, amb l'ordre “tail -f”.

Hem realitzat una investigació que ha portat hores i al final hem descobert que el problema era degut a una opció de NFS situada a /etc/exports del node màster.

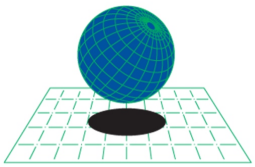
La opció per defecte era la de mantenir les transferències asíncrones per tal de millorar el rendiment del sistema de fitxers.

Aquesta opció feia que els canvis fossin escrits a disc només quan passés un d'aquests events:

- Es demana forçosament més memòria del sistema
- Una aplicació fa un *“flush”* de fitxers explícitament amb la crida sync, msync o fsync.
- Una aplicació tanca un fitxer amb close.
- El fitxer és bloquejat o desbloquejat amb fcntl.

En resum, un sistema NFS muntat amb la opció async podía fer que les dades escrites per l'aplicació no fossin immediatament vistes pel servidor.

La solució passa per activar l'opció sync que permet tenir una gran coherència de dades però amb un impacte major en el rendiment, fent que cada crida al sistema que escriu dades a un fitxer causi immediatament un *flush* de dades cap al servidor NFS.



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 7
Eines de suport i documentació

7.1 Objectius de la documentació

La infraestructura que hem muntat per portar a terme la documentació del projecte s'orienta a tres tipus de receptors. En un primer lloc els usuaris finals que faran ús del servei i que el seu objectiu és calcular el més ràpid i eficaçment possible, en un segon lloc els gestors del servei que seran els encarregats de fer altes, baixes, modificacions, revisions, etc. i finalment en tercer lloc els enginyers que estaran capacitats per implementar o realitzar modificacions importants del sistema.

La documentació pretén explicar com es realitzen les tasques habituals de cadascún d'aquests grups.

7.2 Solucions de documentació implementades

7.2.1 Documentació d'usuari

Té com a objectiu informar als usuaris de tots els aspectes d'usabilitat del servei de càlcul, des de com han de fer servir el node màster, com utilitzar certes eines, consells de programació o enllaços a documentació externa, etc. i també comunicar novetats o noticies importants als usuaris.

Per això hem determinat que el millor sería centralitzar tota aquesta informació en una pàgina web accessible des de Internet.

La pàgina l'hem allotjada al servidor web de CIMNE sota l'adreça *http://hpc.cimne.upc.edu*.

El sistema instal·lat és un Joomla 2.0 que podrà ser administrat per els administradors de sistemes CIMNE i contindrà notícies, informació del hardware i guies bàsiques de l'ús del clúster. També hi haurà enllaços a documentació externa més extensa, com la de RHEL 6.x o SLURM, Figura 82.

Figura 82: Pàgina web http://hpc.cimne.upc.edu

Per altra banda i per agilitzar la comunicació entre usuaris, i per poder emetre comunicats que arribin amb urgència a tothom, s'ha decidit crear una llista de correu. S'aprofitarà el sistema d'altes d'usuari que disposa CIMNE que ja recull la informació de l'usuari per disposar de les adreces d'aquests i es crearà la nova llista anomenada “CIMNE HPC” al servidor de correu de Sistemes. L'enviament a la llista s'efectuarà a *hpc@listas.cimne.upc.edu* i la gestió serà duita a terme pels administradors de Sistemes CIMNE.

Es poden consultar els arxius de la llista disponibles només pels membres registrats a la pàgina web: *http://listas.cimne.upc.edu/cgi-bin/mailman/private/hpc/* , Figura 83.

December 2011 Archives by thread

- **Messages sorted by:** [subject] [author] [date]
- **More info on this list...**

Starting: *Fri Dec 9 13:25:37 CET 2011*
Ending: *Tue Dec 27 10:05:41 CET 2011*
Messages: 11

- [CIMNE-HPC] Crash pez013 *Felip Moll*
- [CIMNE-HPC] Reserva de mantenimiento *Felip Moll*
  - [CIMNE-HPC] consulta sobre uso del cluster *Julio Marcelo Marti*
    - [CIMNE-HPC] consulta sobre uso del cluster *Felip Moll Marquès*
- [CIMNE-HPC] tabla con toda la info de los nodos *Antonia Larese De Tetto*
  - [CIMNE-HPC] tabla con toda la info de los nodos *Felip Moll*
    - [CIMNE-HPC] tabla con toda la info de los nodos *Miguel A. Pasenau de Riera*
      - [CIMNE-HPC] tabla con toda la info de los nodos *Felip Moll*
- [CIMNE-HPC] Problemas electricos *Felip Moll Marquès*
  - [CIMNE-HPC] Problemas electricos *Antonia Larese De Tetto*
    - [CIMNE-HPC] Problemas electricos *jlozano at cimne.upc.edu*

Last message date: *Tue Dec 27 10:05:41 CET 2011*
Archived on: *Tue Dec 27 10:05:42 CET 2011*

- **Messages sorted by:** [subject] [author] [date]
- **More info on this list...**

This archive was generated by Pipermail 0.09 (Mailman edition).

Figura 83: Arxiu de Desembre de 2011 - Llista HPC CIMNE

7.2.2 Documentació del gestor

Aquesta documentació té com a objecte mostrar les tasques bàsiques de gestió com les d'afegir, modificar i eliminar usuaris del sistema, com modificar les quotes, etc. S'ha documentat seguint l'esquema que de Sistemes CIMNE a la seva wiki interna *https://wikiSistemes.cimne.upc.edu* , sota el servei “Serveis de càlcul”.

7.2.3 Documentació d'administrador

La documentació d'administrador preten mostrar com s'ha implementat el sistema donant tots els coneixements necessaris per realitzar qualsevol tasca d'instal·lació, actualització, etc. Haurà de ser complementada amb les dues altres documentacions esmentades en aquest capítol.

Podem trobar-la a la wiki de Sistemes CIMNE seguint l'esquema que el departament ha implementat. També en forma part aquest mateix document i totes les referències bibliogràfiques que es mencionen.

7.3 Eines de suport

Mencionem a continuació quines són les principals eines que hem implementat per tal de realitzar un monitoreig de la càrrega del sistema, dels processos en execució i de l'estat del sistema en general.

7.3.1 Ganglia

Accessible des del servidor web instal·lat al node màster ens dona la possibilitat de veure l'estat del clúster en general i de cada un dels seus nodes per separat. Podem veure informació com la càrrega global del sistema, la càrrega de xarxa, numero de processos, etc.

Per accedir-hi podem fer-ho públicament a l'adreça:

http://acuario.cimne.upc.edu/ganglia

7.3.2 Comandes SLURM

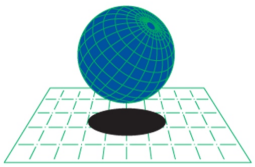
Diversos usuaris ens han demanat la forma d'obtenir certes informacions sobre els seus treballs o sobre com modificar-los i controlar-los. A continuació posem la llista d'eines més importants que poden utilitzar.

- sbatch – enviar un treball en format script a SLURM
- squeue – veure els treballs actuals a la cua
- sinfo – veure quines i quin és l'estat actual de les particions
- scancel – cancelar un treball
- sview – executar l'eina gràfica per controlar treballs, particions i estat dels nodes
- smap – executa l'eina de cònsola per controlar treballs, particions i estat dels nodes
- sacct – mostra dades estadístiques dels treballs com memòria mitja consumida, cpu, etc.
- sshare – mostra informació sobre l'estat del FairShare
- scontrol – permet modificar treballs en cua
- sstat – mostra informació detallada sobre els treballs actualment en execució
- sreport – mostra informació de la utilització del clúster i els recursos al llarg del temps

7.3.3 Informació del sistema

És freqüent rebre consultes sobre el hardware disponible al sistema. Encara que aquest estigui detallat a la pàgina web *http://hpc.cimne.upc.edu* pot ser util utilitzar les comandes següents:

- ibstat – ens dona l'estat de l'enllaç infiniband
- lstopo – mostra l'arquitectura del sistema actual, els nivells de caché, nuclis, numa, etc.
- /proc/cpuinfo – mostra les instruccions del processador, línies de caché, etc.
- /proc/meminfo – mostra l'ús actual de la memòria
- dmidecode – obté informació detallada de la memòria, busos PCI i cpus
- lscpu – mostra la informació d'una manera més completa que /proc/cpuinfo
- cpuid – informació obtinguda directament de eax/1, ebx/1, ecx/1, edx/1
- top – activant la visualització SMP, threads i CPU's (tecles f-j, 1, H) mostra on s'allotja cada procés de cada usuari
- cexec – executa una comanda a tots els nodes
- quota – mostra la quota d'un usuari o d'un grup i la utilització actual de l'espai
- ipmitool - mostra la informació del sistema agafada dels sensors



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 8
Estudis de hardware i sostenibilitat

8.1 Estudi de l'arquitectura de hardware i implicacions

Aquest estudi pretén donar una visió global de l'arquitectura de hardware de la que disposem als nodes de càlcul del clúster. També veurem quines implicacions generals tenen en la programació dels codis els diferents dissenys analitzats.

Al disposar de tres tipus de nodes i per tant tres arquitectures diferents dividirem l'estudi en també tres apartats.

8.1.1 Intel® Xeon® E5410 (Nodes M600)

Aquest processador prové de la modificació d'arquitectura que encongí el xip de l'anterior micro-arquitectura Intel Core, nom clau Merom, de 65nm a 45nm (cicle “tick” de Intel). El nom clau de la nova micro-arquitectura és Penryn (no confondre amb el nom clau del processador Penryn per portàtils) i fou llançada l'11 de Novembre de 2007. D'aquesta micro-arquitectura sorgiren quatre branques anomenades Dunnington, Harpertown, Yorkfield i Wolfdale, per servidors de 8 o 4 processadors, ordinadors d'escriptori de 4 i 2 processadors, ordinadors d'alt rendiment i ordinadors d'escriptori respectivament, [58].

L'Intel Xeon E5410 correspon al nom comercial del processador Harpertown, CPUID 010676h. Per estudiar les seves característiques ens centrarem en general amb la micro-arquitectura Penryn tenint en compte les particularitats del Harpertown.

Micro-arquitectura

Es tracta d'un processador segmentat de 64 bits que treballa a una freqüència interna màxima de 2.33Ghz, compta amb 820 milions de transistors i utilitza una tecnologia de fabricació de 45nm.

Disposa de 14 etapes que permeten realitzar execucions d'instruccions fora d'ordre.

A la Figura 84 veiem en detall quines són aquestes etapes i en comentem el flux general:

Començant per l'etapa de “fetch” i pre-descodificació veiem com el ITLB de 128 línies suporta una caché associativa de 8 vies i 32KB per instruccions . Aquestes instruccions són posteriorment pre-descodificades i són convertides en 6 instruccions que es descodifiquen completament als descodificadors simples i al complex.

Una vegada descodificades entren en un buffer i són enviades al RAT i al buffer de re-ordenament que permet pausar algunes instruccions (Retirement Register File) mentre s'estan executant altres fora d'ordre, esperant algun resultat o dades de E/S.

Posteriorment a la re-ordenació entren a l'estació de reserva i a continuació es realitza la operació en les diferents unitats funcionals, ja siguin les dues ALU senzilles, les de SSE, les d'operació de coma flotant, les d'operació a memòria, les que van a la unitat funcional ALU Branch, etc.

Precisament aquesta última unitat funcional, la ALU Branch és una introducció de nova tecnologia en aquests processadors i es tracta de la fusió de Macro-operacions realitzada pel bloc “Complex Decoder” que combina dues instruccions x86 amb una sola micro-operació, per exemple una comparació seguida d'un salt condicional es convertiria amb una sola i s'executaria a l'“ALU Branch”.

Finalment veiem com hi ha accés a la caché de nivell 1 8-associativa de 32KB, amb una TLB de dades de 16 línies i connectada a la caché de nivell 2 compartida amb un altre nucli. Aquesta última connectada al FSB per accedir a MP.

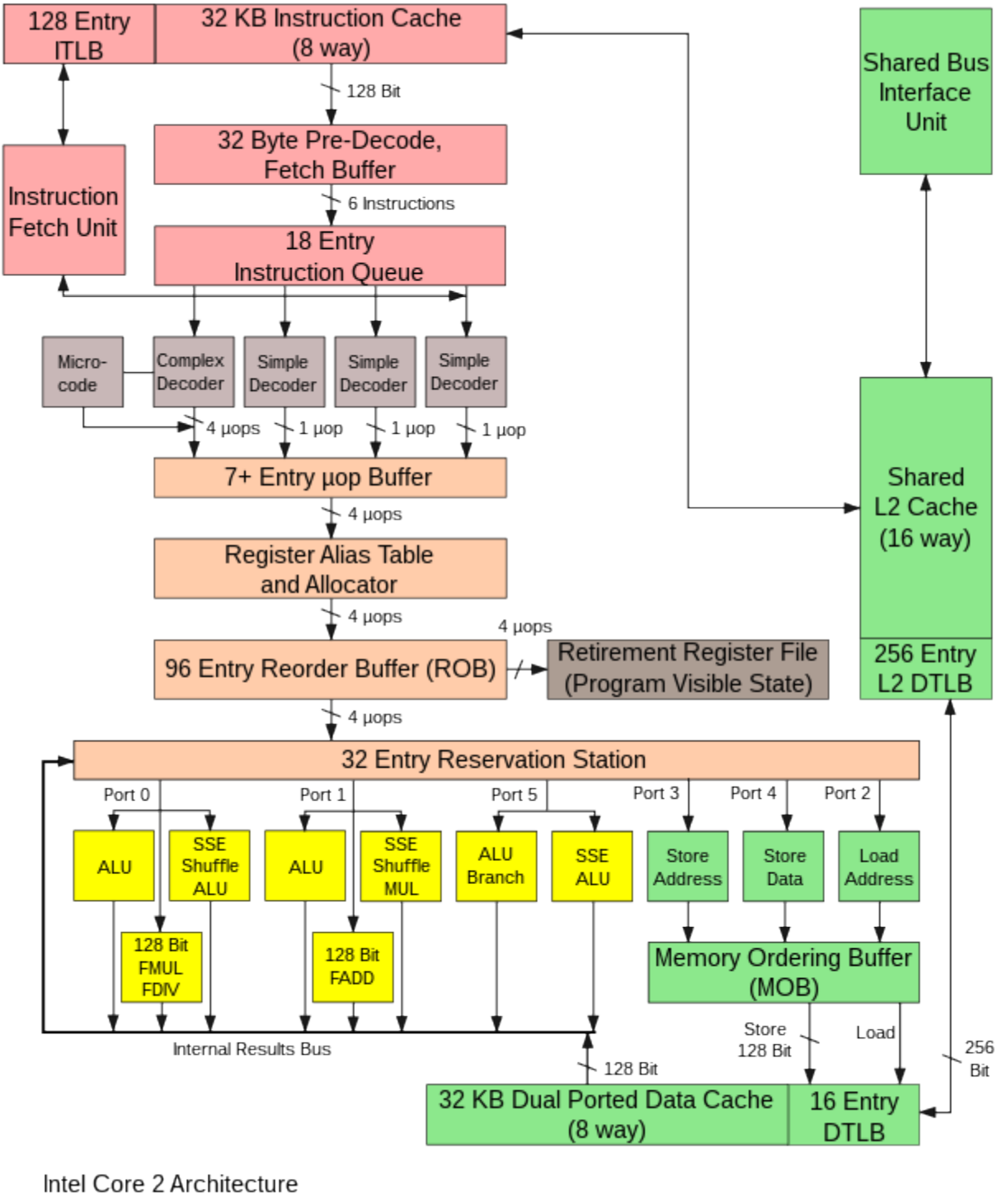


Figura 84: Micro-arquitectura d'un nucli Intel Core2 (Penryn) [59]

El nombre de cicles total mesurat per la latència d'una operació de fall de predicció en un salt és de 14 cicles.

Les característiques més rellevants de la micro-arquitectura Core de 45nm en relació a les anteriors són la macro-fusió d'operacions i les instruccions SSE4.

Per comentar més en detall la macro-fusió d'operacions mostrem amb un exemple quina avantatge es pot treure.

A la part dreta de la Figura 85, disposem d'una seqüència d'instruccions que entren al buffer de descodificació. Actuant sense macro-fusió veiem com ens arriben 5 instruccions per descodificar i ho fem d'una en una tardant en total dos cicles ja que només tenim 4 descodificadors.

A la part dreta, implementant la macro-fusió, aprofitem el bloc de descodificació complex que ens permet fusionar la instrucció de comparació i la de salt en una sola i llavors descodifiquem les instruccions restants. Tot amb un sol cicle.

Amb aquesta tècnica i afegint el descodificador complex es pot arribar a aconseguir una millora del 66% en aquesta etapa [60].

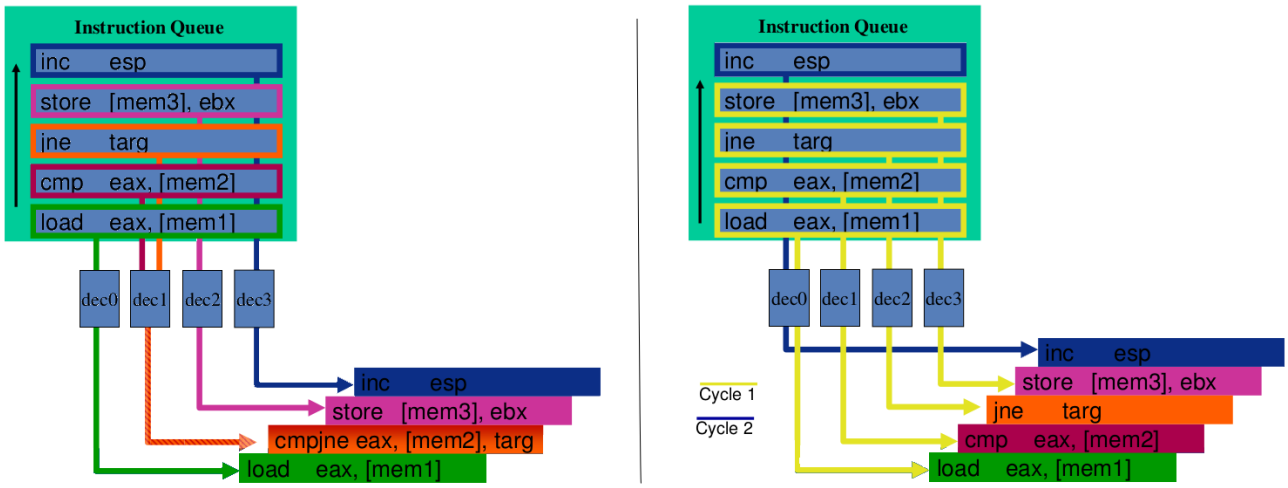


Figura 85: Descodificació sense macro-fusió (esquerra) i amb macro-fusió (dreta)

En el cas de les instruccions SSE4.2 hem de comentar que són instruccions de vectorització SIMD i que operen en registres de 128 bits. Aquests registres poden contindre diferents tipus d'informació com mostrem a la Figura 86.

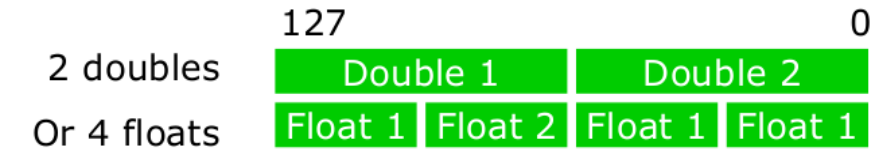


Figura 86: Dades contingudes a una instruccio SSE

La millora implementada en una arquitectura Core obté un rendiment d'un cicle per totes les instruccions SSE de 128-bits, Figura 87.

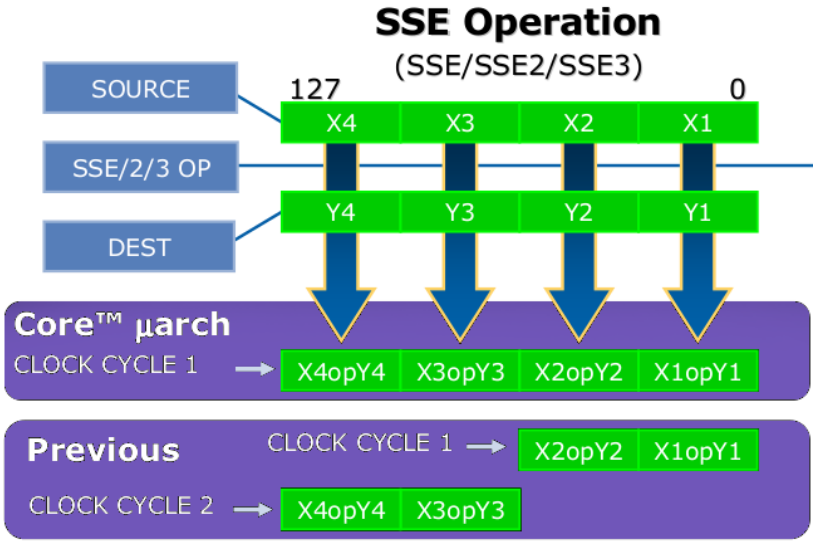


Figura 87: Comparació de la latència d'1 cicle per SSEx a un Intel Core vs anteriors micro-arquitectures.

La micro-arquitectura Core al ser descendent de la P6, no disposa de la tecnologia Hyper-Threading però si d'instruccions de virtualització Intel VT-x, Intel 64 i SSE4.

Incorpora també la NX bit (anomenada per Intel XD bit) que permet definir zones de memòria que no s'executaran mai i que serviran per emmagatzematge de dades, instruccions o codi.

Altres instruccions rellevants que admet el processador són (obtingut amb “dmidecode”):

```
FPU (Floating-point unit on-chip)      PAT (Page attribute table)
VME (Virtual mode extension)           PSE-36 (36-bit page size ext)
DE (Debugging extension)               CLFSH (CLFLUSH instruction)
PSE (Page size extension)              DS (Debug store)
TSC (Time stamp counter)               ACPI (ACPI supported)
MSR (Model specific registers)         MMX (MMX technology supported)
PAE (Physical address extension)       FXSR (Fast floating-p sv&restore)
MCE (Machine check exception)          SSE (Streaming SIMD extensions)
CX8 (CMPXCHG8 instruction)             SSE2 (Streaming SIMD extens 2)
APIC (On-chip APIC hardware)           SSE3 (Streaming SIMD extens 3)
SEP (Fast system call)                 SSE4.1 (Streaming SIMD extens 4)
MTRR (Memory type range registrs)      SS (Self-snoop)
PGE (Page global enable)               HTT (Hyper-threading technology)
MCA (Machine check architecture)       TM (Thermal monitor supported)
CMOV (Conditional mov instruct)        PBE (Pending break enabled)
```

Finalment s'introduí una millora en l'eficiència energètica implementant la tecnologia anomenada per Intel com a Demand Based Switching coneguda també com Speed Step, que permet ajustar la freqüència del processador en funció de la utilització.

Macro-arquitectura

Els nodes M600 disposen d'una placa base amb el xip controlador de memòria Intel® 5000P, nom clau Blackford.

El MCH (Memory Controller Hub o northbridge) 5000P fou utilitzat primer de tot per els Pentiu M el 2006 i es re-utilitzà fins el 2007. Aquest tenia problemes d'E/S que solucionava el nou Seaburg 5400, llançat el 2007. Malauradament pocs fabricants utilitzaren el Seaburg (tampoc Dell) tot i que era millor en rendiment. Un exemple on si s'utilitzà foren els EMC Symmetrix VMAX, sistemes d'emmagatzematge de dades massius, fet que ens fa preguntar-nos si el 5000P no era suficientment bo per l'E/S que requerien aquells sistemes.

Al poc temps, l'aparició de noves arquitectures NUMA com el Nehalem deixà de banda els xips northbridge i els processadors Penryn foren els últims a portar FSB i MCH.

L'esquema d'un node com l'M600 agafa la forma de la Figura 88 tot i que amb el 5000P, [61].

Es tracta d'un sistema Core2 de dues vies de FSB associada cadascuna a un socket. A cada socket hi ha un processador que conté dos xips, cadascun amb 2 nuclis bàsics. En total formen dos processadors de quatre nuclis.

El fet d'haver-hi dos xips de dos nuclis dins cada processador ho permet el protocol de FSB que permet que múltiples nuclis comparteixin un mateix bus.

Cada xip (cada dos nuclis) comparteix memòria caché de nivell 2.

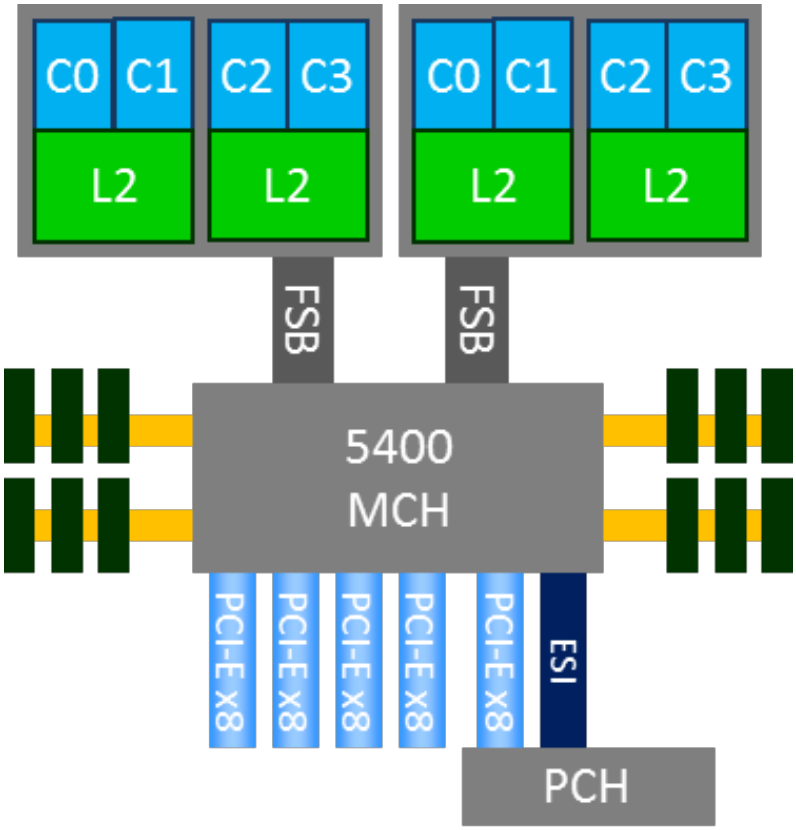


Figura 88: Arquitectura del node M600

Per assegurar-nos de que l'arquitectura plantejada és correcta, hem executat la comanda “lstopo” que fa una crida a “hwloc”, utilitat que analitza el hardware del sistema i el presenta de forma elegant.

A Figura 89 podem veure la sortida de la comanda “lstopo”, un resum de les cachés i nuclis físics dels que disposa un node M600, concretament Acuario.

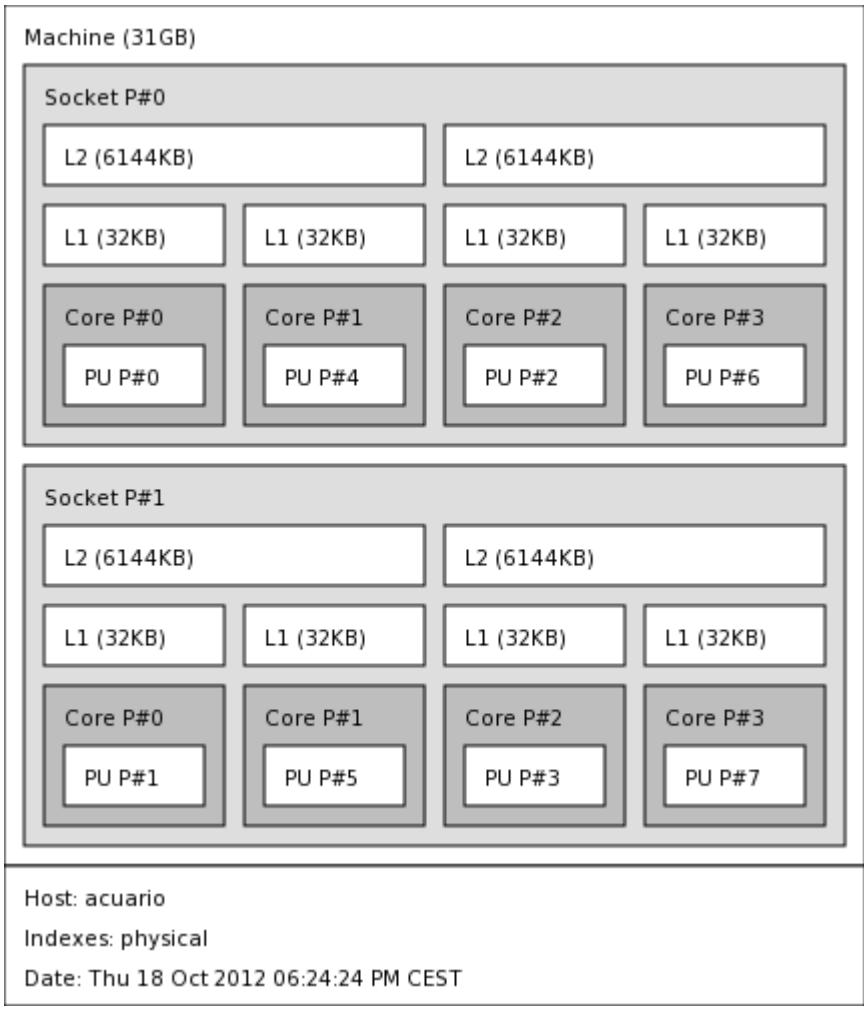


Figura 89: Sortida de lstopo usant hwloc al node Acuario

Per altra banda el MCH 5000P disposa de 4 canals de memòria, 24 enllaços PCI-E i un ESI, que és essencialment un port PCI-E x4 amb compatibilitat per funcions més antigues. S'anomena DMI en un sistema d'escriptori convencional.

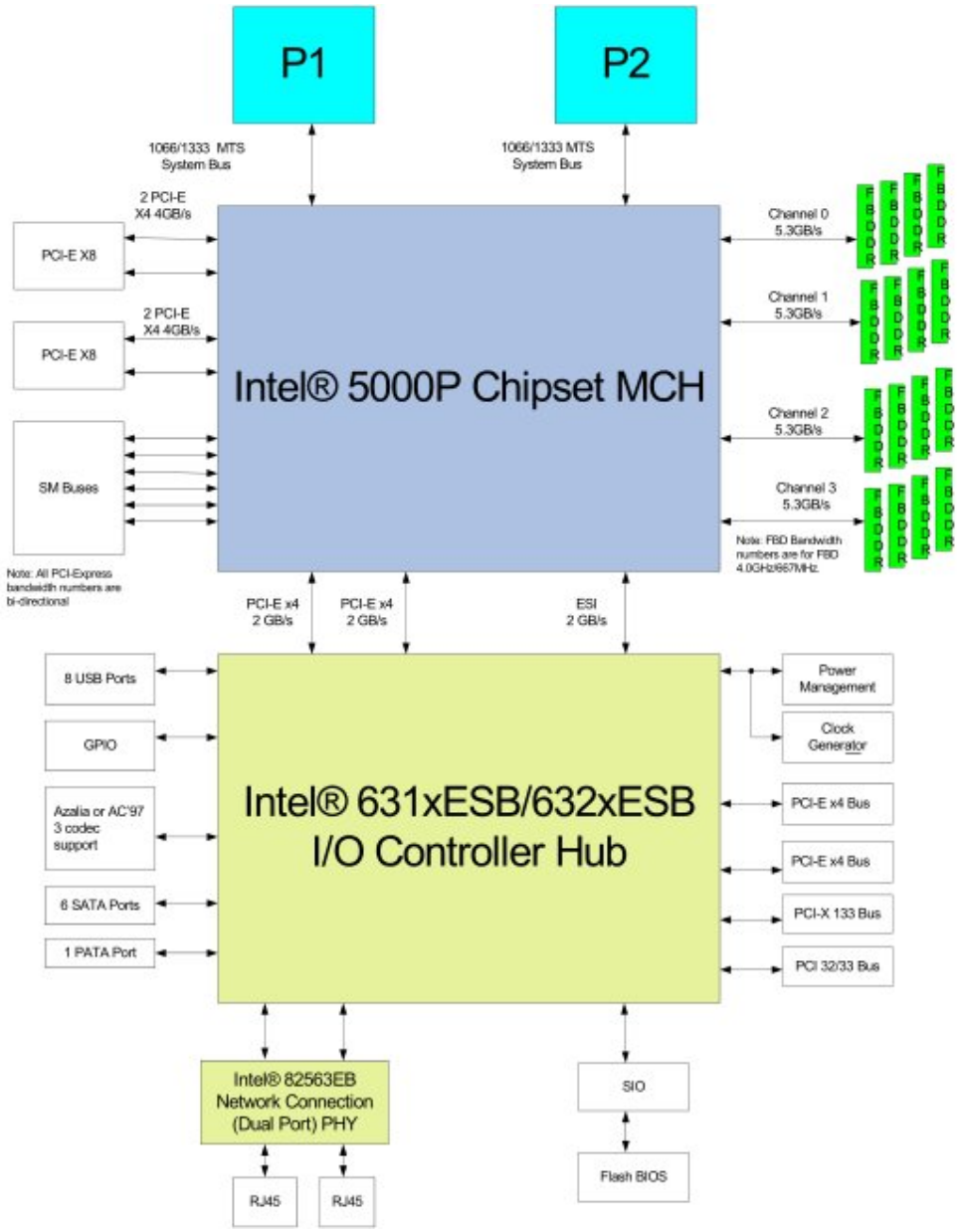


Figura 90: Diagrama del xip 5000P [62]

A l'M600 Cada FSB funciona a una freqüència màxima de 1333Mhz.

Cada canal de memòria pot suportar fins a 4 mòduls FB-DIMM DDR2. Els canals poden ser organitzats en dues branques que suportin RAID 1 (mirroring) i es poden instal·lar un total de 16 DIMM amb un màxim de 64GB en mode normal o 32 GB en mode RAID 1.

L'ample de banda de lectura de cada canal FB-DIMM és de 5.3GB/s que dona un total de fins a 21 GB/s pels quatre canals sumats. L'ample de banda d'escriptura és de 10.7GB/s.

Les característiques de la memòria FB-DIMM l'hem comentada anteriorment en aquest document al capítol 4.1.2.2.

Energia

Els fabricants de processadors habitualment fan servir un nombre de “steppings” que representen millores incrementals i funcionalitats afegides com les mides de caché i control d'energia.

La majoria d'aquests “steps” es modifiquen entre branques de processadors d'una mateix arquitectura típicament desactivant algunes característiques o limitant les freqüències de rellotge. Així s'aconsegueix que amb una mateixa arquitectura es disposin de processadors de diferents games.

L'”stepping” del nostre E5410 es pot obtenir amb la informació de “dmidecode” o /proc/cpuinfo. Veiem d'aquesta manera que el CPUID és “010676h” i l'”step” és el 6 revisió C0. Hi ha un altre E5410 amb stepping E0 que millora l'eficiència energètica afegint dues noves instruccions: XSAVE i XRSTOR. Fou llançat a l'Agost de 2008 i no l'hem de confondre amb el que disposem actualment.

Les funcionalitats d'energia d'aquest processador milloren les arquitectures anteriors. Incorporen la tecnologia Intel SpeedStep, tècnica coneguda com “demand-based switching” (DBS). Aquesta tecnologia permet modificar dinàmicament el voltatge de la CPU, i per tant la freqüència de treball, en funció de la càrrega del sistema.

D'aquesta manera permet que els processadors consumeixin menys energia i dissipin menys calor.

La funcionalitat DBS s'inclou al kernel de Linux i és pot controlar mitjançant els dispositius de /sys/devices/system/cpu/cpu*/cpufreq/ i l'eina cpufreq.

El consum dels processadors es situa en un màxim de 80W quan treballen a màxima freqüència segons les especificacions oficials d'Intel.

8.1.2 AMD Opteron™ 2356 (Nodes M605)

Aquest processador es tracta d'un AMD K10 (o 10h) d'arquitectura Opteron fabricat amb tecnologia de 65nm i llançat el setembre de 2007. Fou el primer de la sèrie K10.

El mateix any de llançament AMD va parar les ventes d'aquest processador degut a un bug a la TLB a l'"stepping" B2 que podia causar un "race condition" i bloquejar el sistema. Es van treure funcionalitats per permetre a la BIOS desactivar el TLB i es van crear "patches" pel Kernel de Linux que solucionaven el problema, però amb un 5 a 20% de penalització. El 2008 AMD va solucionar el problema i va sortir el nou "stepping" B3. No hem estat capaços de determinar quin "stepping" de processador disposem exactament tot i la informació proporcionada per "dmidecode", /proc/cpuinfo i "lscpu". El CPUID és 100F23h.

El nom comercial del processador és Barcelona, començant amb una sèrie de noms dels circuits de Formula 1 mundials, [63].

A diferència de l'Intel E5410 aquest processador CISC és d'arquitectura NUMA.

Micro-arquitectura

El processador és molt semblant en micro-arquitectura al Xeon E5410. De fet incorpora algunes funcionalitats que Intel va emprar per millorar el rendiment per tic de rellotge. A simple vista podríem dir que l'arquitectura Core2 és un 33% més ample que el Barcelona, però en codi real el rendiment és molt similar.

Els nuclis funcionen a una freqüència màxima de 2300Mhz, una mica inferior que els E5410.

A la Figura 91 mostrem el diagrama de blocs de la micro-arquitectura del Barcelona. Comentarem a continuació les característiques principals i les diferencies importants respecte al Core2.

Començant per la fase de "fetch" d'instruccions veiem que encara que Barcelona disposi d'una ITLB més petita amb 48 línies, les instruccions són agafades en conjunts 256 bits cada cicle mentre que en el Core2 ho feien amb 128 bits/cicle. Aquest ample de bus major dona un avantatge amb les instruccions grans com les SIMD o les de 64 bits que en aquest processador seran adquirides més ràpidament. Com a conseqüència s'ha incrementat el buffer de pre-decode a més de 32B (diferent en funció de l'"stepping").

Ser més eficaç en adquirir instruccions de 64 bits o SIMD fa al Barcelona un nucli ideal per tasques de HPC ja que aquestes instruccions són freqüents.

En aquesta mateixa etapa s'inclou el predictor de salts. Disposa d'un registre històric global de salts que emmagatzema els últims 12 salts. Incorpora també un predictor de salts indirectes especialment dissenyat per salts amb diversos destins, com en el cas dels "switch".

Com al Core2 es disposa d'un pre-descodificador que analitza la mida d'una instrucció i en marca el principi i el final. No disposa però de cap unitat que permeti fusionar macro-instruccions, encara que les instruccions de 128 bits SSE són descodificades a una sola micro-operació, fet que fa que el mecanisme de fora d'ordre i estacions de reserva sigui més efectiu.

El mecanisme de fora d'ordre és més complicat que al Core2 ja que s'ha dividit la part d'operacions de coma flotant i d'enters, mentre que el Core2 tenia una sola unitat per controlar-ho tot. Aquest és un dels punts forts de AMD respecte les operacions de coma flotant.

Un dels canvis importants del Barcelona és que la unitat entera de divisió, IDIV, és de latència variable en funció dels operands: 23 cicles + el nombre de bits significatius en el valor absolut del dividend. Això fa que operacions IDIV de enters sense signe siguin 10 cicles més ràpides.

Finalment una diferència important d'AMD vs Intel és que el primer separa la unitat de generació d'adreços (AGU) de les unitats de "load" i "store" (LSU), fet que implica més unitats al xip i per tant més dissipació de calor, però millor rendiment en certs casos.

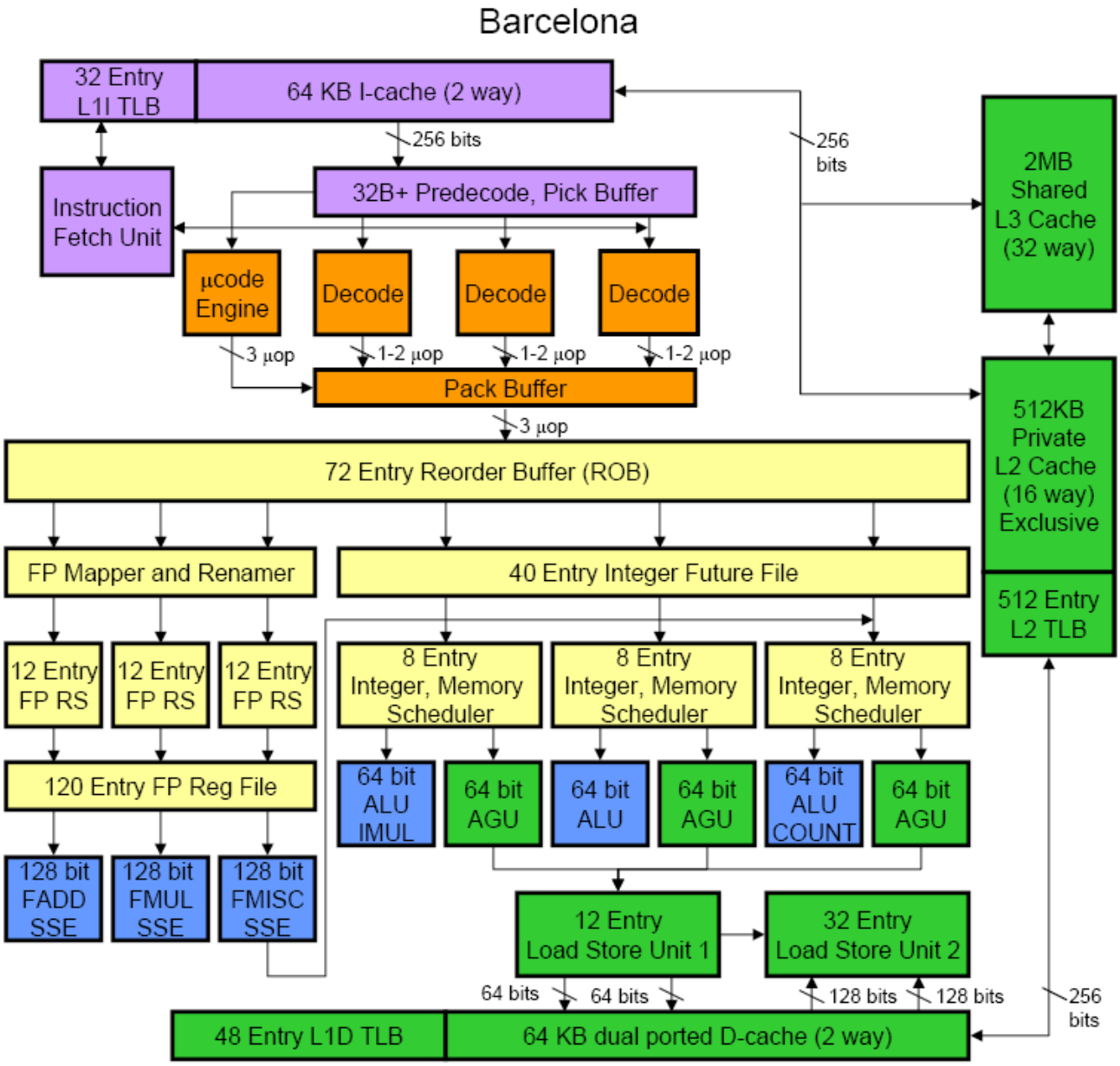


Figura 91: Diagrama de blocs de la micro-arquitectura Barcelona [64]

Per acabar amb aquest resum veiem que la forma d'accedir a memòria és diferent del Core2. Primer de tot veiem que disposem d'una caché de nivell 1 del doble que la del E5410, i a més un TLB també més gran. Per altra part disposem d'accés a una caché de nivell 2 de 512KB no compartida.

En el Barcelona s'afegeix un nou nivell compartit amb tots els nuclis de 2MB de capacitat.

En quant a conjunt d'instruccions implementa les mateixes que el Core2 excepte per la POPCOUNT, que compta el nombre de 1 o 0's en un registre.

Macro-arquitectura

L'arquitectura dels nodes M605 és de tipus NUMA a diferència dels M600 que disposaven de FSB.

El controlador de memòria s'integra en cada processadors i s'introdueix la tecnologia Hyper-Transport (QPI a Intel) per realitzar la comunicació punt a punt entre processadors i dispositius d'E/S.

El major avantatge d'aquesta arquitectura és que evita els colls d'ampolla del FSB tot i que requereix un mecanisme per mantenir les cachés i la memòria principal en un estat coherent.

Cada controlador de memòria suporta transaccions independents de 64 Bytes i suporta “mirroring” de memòria RAM (RAID 1). Els bancs DDR2 dels que disposa el node M600 fan transferències a 32 Bytes i per tant millora l'eficiència en aquest aspecte.

El Barcelona incorpora també un predictor de pàgines de memòria similar al predictor de salts que determina quines posicions seran accedides i permet carregar o descarregar pàgines en caché, ajudant a mantenir la coherència.

Afageix també quatre busos HyperTransport, tot i que en el nostre sistema només n'utilitzem dos (tenim 2 sockets), fet que redueix la latència ja que cada processador és accessible amb un sol salt. Cada link de HT funciona a 2GT/s, però al ser compatibles amb HT 3.0 podrien operar a 5.2GT/s. A més HT3.0 pot modular l'ample de l'enllaç i la freqüència per estalviar energia.

Per mantenir la coherència a la caché, l'anterior K8 tenia que monitoritzar tot el tràfic entre processadors i memòria del sistema per veure si les dades de les que cada nucli disposava eren coherents amb la resta. Ho aconseguia mitjançant un mecanisme de pregunta-resposta que feia esperar a tots els processadors mentre s'executava la pregunta i s'obtenia la resposta.

En el Barcelona es fan servir els estats M (modified) i O (owned) de les línies, que signifiquen que la memòria té una copia modificada i per tant la CPU no ha d'esperar les respostes dels altres nuclis, [64].

A la Figura 92 i Figura 93 veiem un resum del que acabem d'explicar.

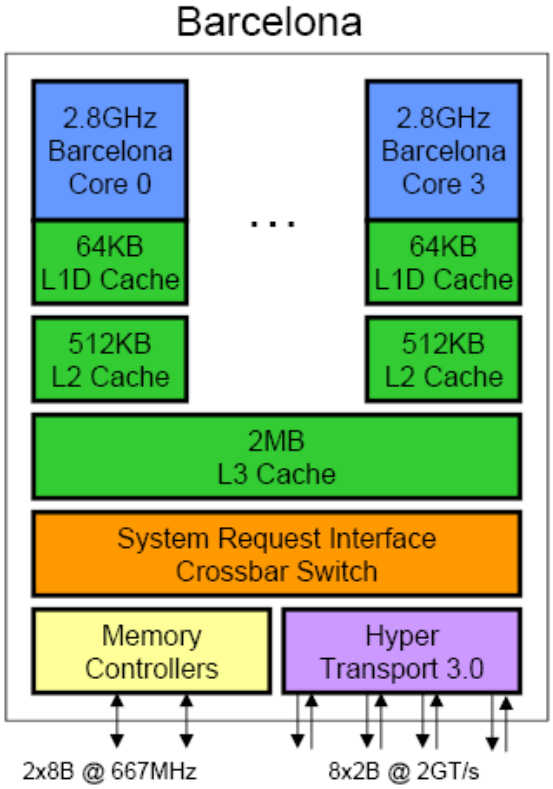


Figura 92: Arquitectura Barcelona

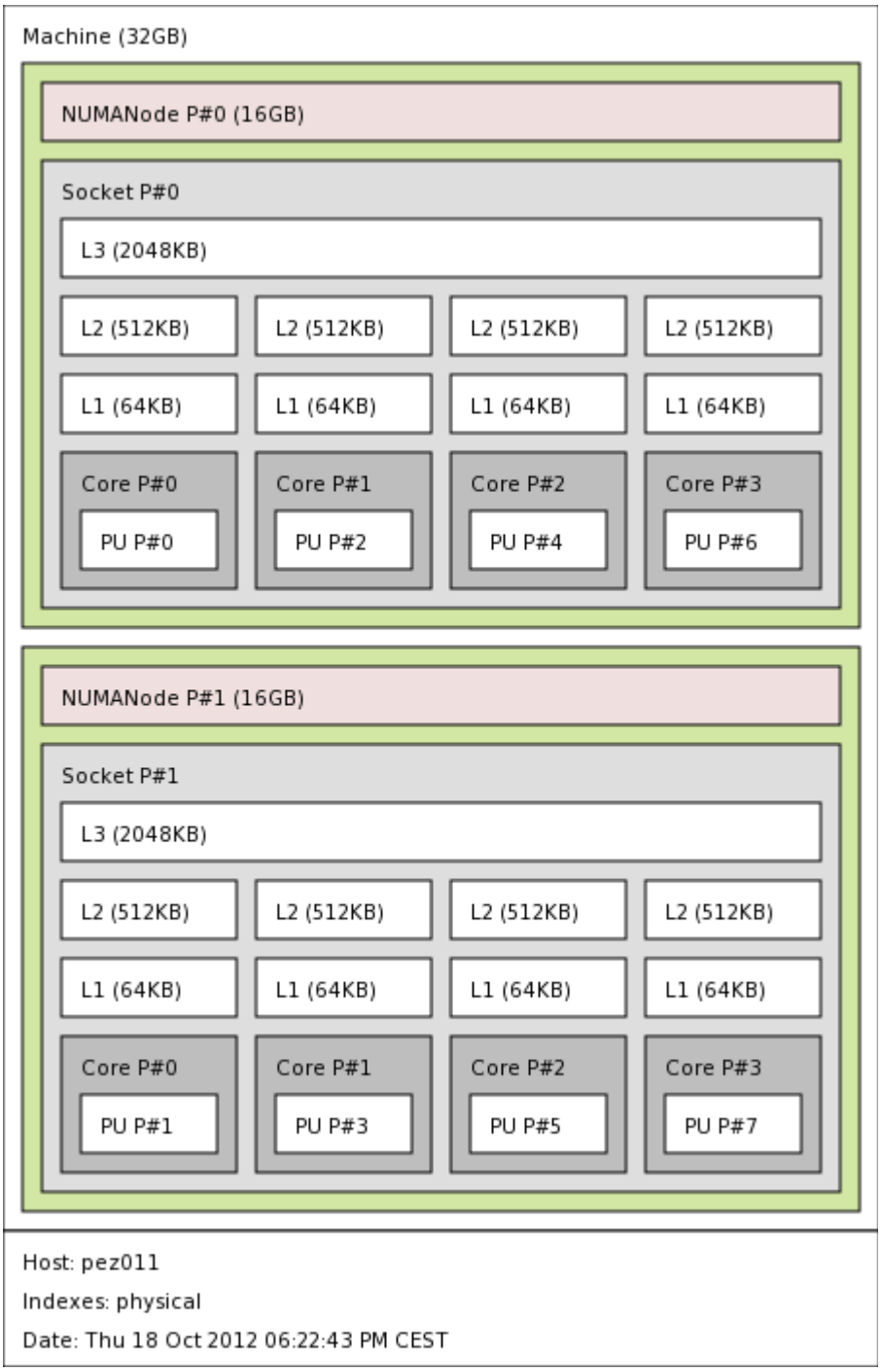


Figura 93: Arquitectura NUMA dels M605, obtingut amb "lstopo"

Energia

Barcelona funciona amb voltatges de 0.8 – 1.4v. Cada nucli és independent dels altres i pot funcionar a diferentes freqüències. Els nuclis són independents de la part compartida (L3, etc) fet que permet gestionar millor els voltatges.

Cada nucli a més disposa de 8 sensors de temperatura al circuit, i el northbridge de 6.

Tot és controlat per un controlador de temperatura que decideix quins modes d'estalvi d'energia aplicar, el marquem en vermell a la Figura 94.

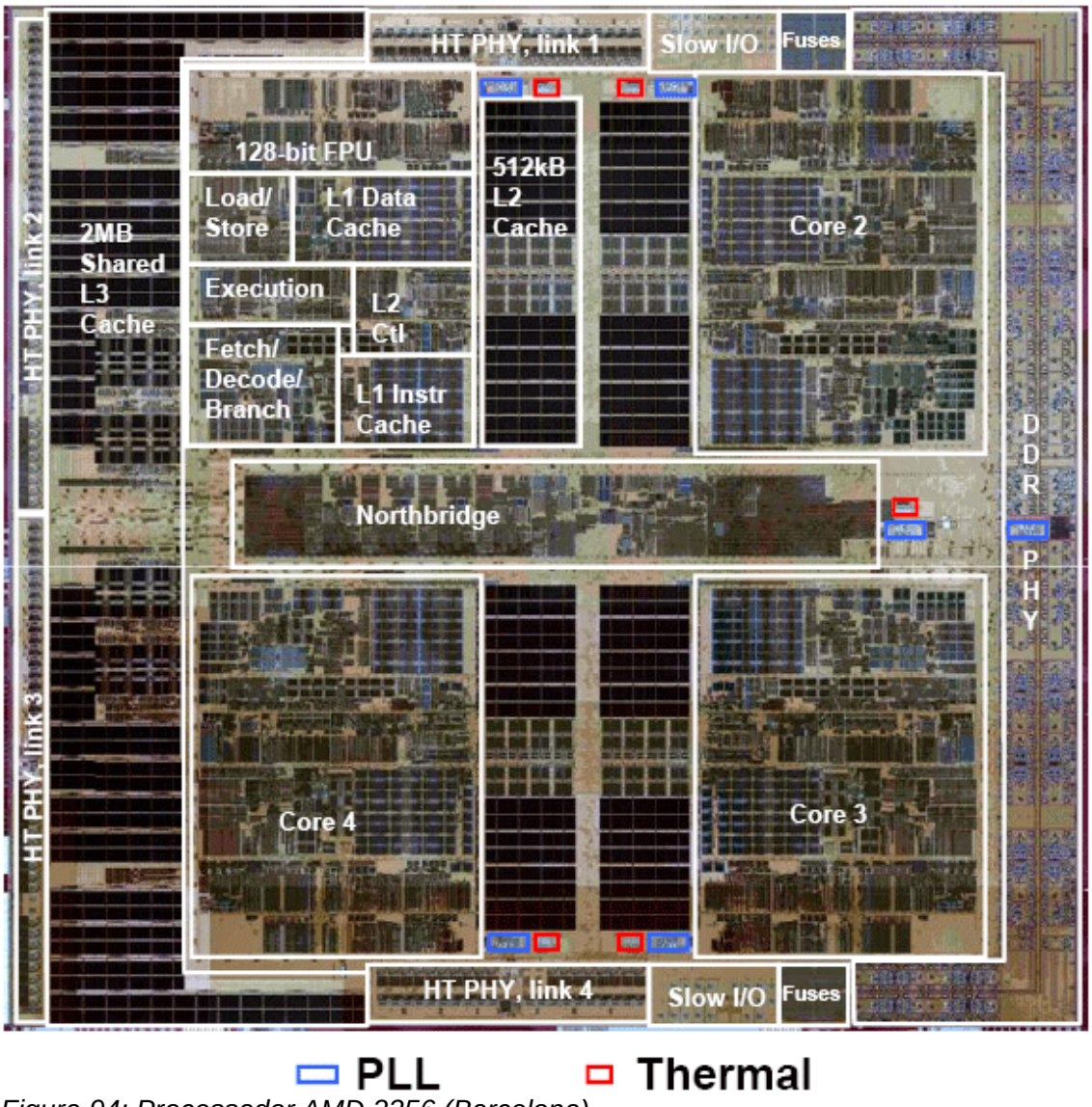


Figura 94: Processador AMD 2356 (Barcelona)

Segons AMD el procesador Opteron 2356 pot dissipar fins a 75W i funcionar a un rang de temperatures d'entre 55 i 76 ºC.

8.1.3 Intel® Xeon® E5645 (Nodes M610)

Aquest processador prové del cicle “tick” de Intel realitzat el 4 de Gener de 2010 modificant la micro-arquitectura Nehalem de 45nm a 32nm, amb el nou nom clau de Westmere.

La micro-arquitectura Westmere compta amb quatre branques de processadors, Westmere-EX, Westmere-EP, Gulftown i Clarkdale per servidors de 8 o 4 processadors, escriptori o servidors de 4 o 2 processadors, escriptori d'alt rendiment i escriptori baix rendiment respectivament.

El processador E5645 correspon al nom comercial del Westmere-EP llançat el 16 de Març de 2010. Nosaltres disposem d'arquitectures de 2 processadors amb 6 nuclis cadascuna.

Micro-arquitectura

Els nuclis del Westmere-EP treballen a 2.4Ghz i poden arribar a 2.67Ghz en mode Turbo.

A diferència del seu predecessor, el Nehalem incorpora de nou la tecnologia Hyper Threading i millora la predicció de salts, factor important per les aplicacions multi-fil.

Aquestes millores són la incorporació d'un predictor de salts de caché de L2 molt més refinat quan es tracta de codis grans (per exemple en bases de dades) i la incorporació d'un RSB (Advanced Renamed Return Stack Buffer) que elimina les prediccions incorrectes en instruccions x86 RET.

La caché de primer nivell i els TLBs es mantenen igual amb 32KB i 128 línies 4-associatives per dades i instruccions, però s'inclou a més una caché TLB de segon nivell unificat de 512 línies pensat per pàgines petites (4K) , a diferència del de només dades de 256 línies i nivell 2 al Penryn.

Disposa de 20 a 24 etapes a diferència del Core2 que en tenia 14.

La micro-arquitectura ha estat adaptada per integrar la tecnologia Hyper Threading, operacions que suporten els Xeon E5645.

S'ha millorat l'operació de fusió de macro-operacions respecte al Penryn per permetre gestionar operacions de 64 bits, fet que fa millorar molt el rendiment en instruccions de mida gran.

Per altre banda s'han millorat totes les unitats, com per exemple la de fora d'ordre o les estacions de reserva, augmentant els buffers a 28 micro-operacions comparades amb les 7 que disposava el Core2.

Els buffers de ”load” i “store” també s'han augmentat de 32 i 20 a 48 i 32 entrades respectivament.

Podem comentar que en general tota la micro-arquitectura ha sigut modificada augmentant els amples de banda, millorant els algorismes i adaptant-la a l'hyper-threading a més d'afegir noves instruccions, com les SSE 4.2 i AVX. [38],[41],[65],Figura 95.

Recomanem especialment llegir el paper “The Architecture of the Nehalem Processor and Nehalem-EP SMP Platforms, M.E.Thomadakis” [41], que escriu un resum molt acurat de l'arquitectura Nehalem-EP (multi-socket).

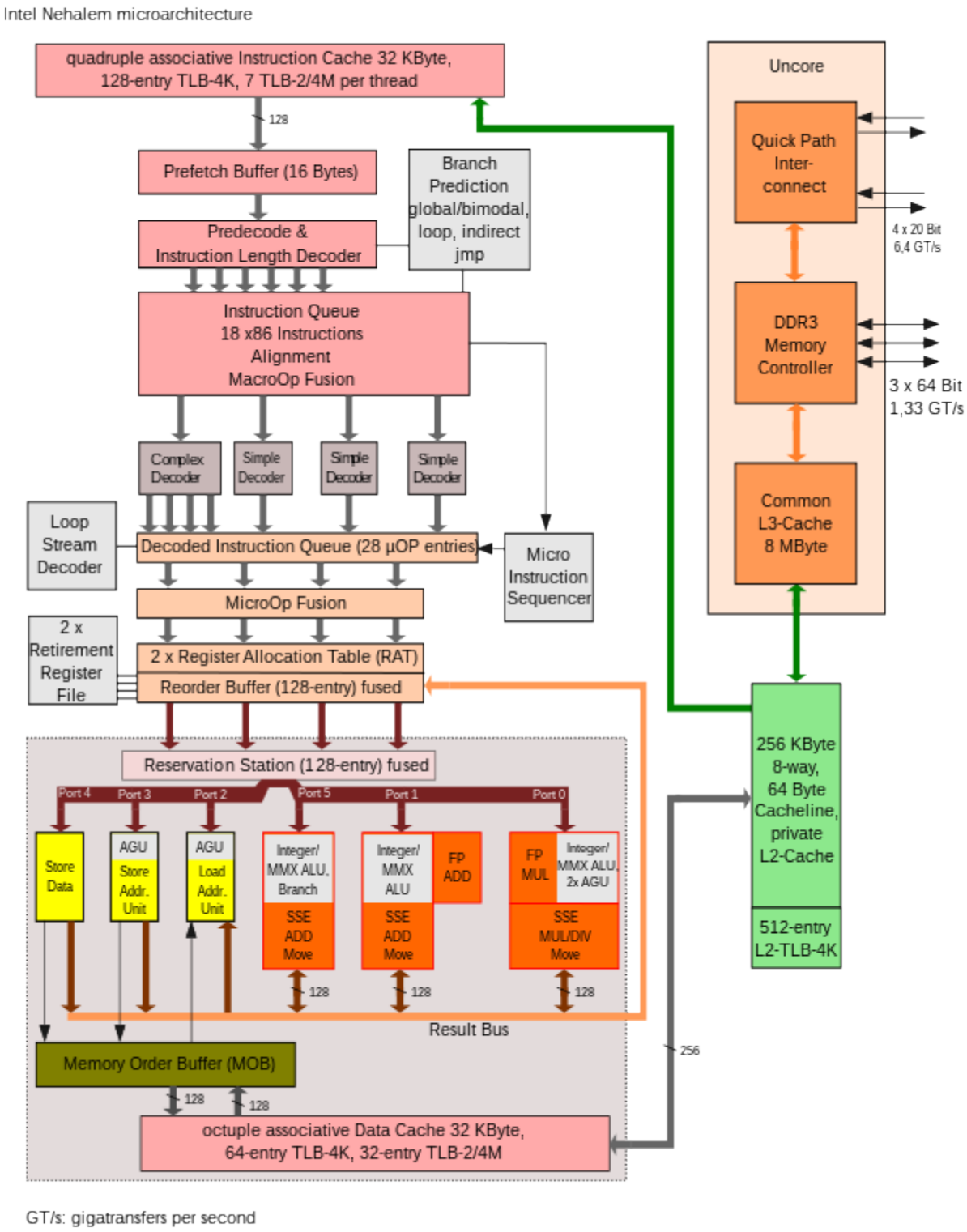


Figura 95: Micro-arquitectura Nehalem

Macro-arquitectura

Els xips Westmere-EP disposen d'un controlador de memòria integrat (IMC) amb canals DDR3 de 8 bytes amb una velocitat de fins a 1.333 GT/s, Figura 96. En total, l'ample de banda de la memòria sumant els 3 canals pot arribar a 32GB/s.

La velocitat dels links QPI és de 5.86GT/s.

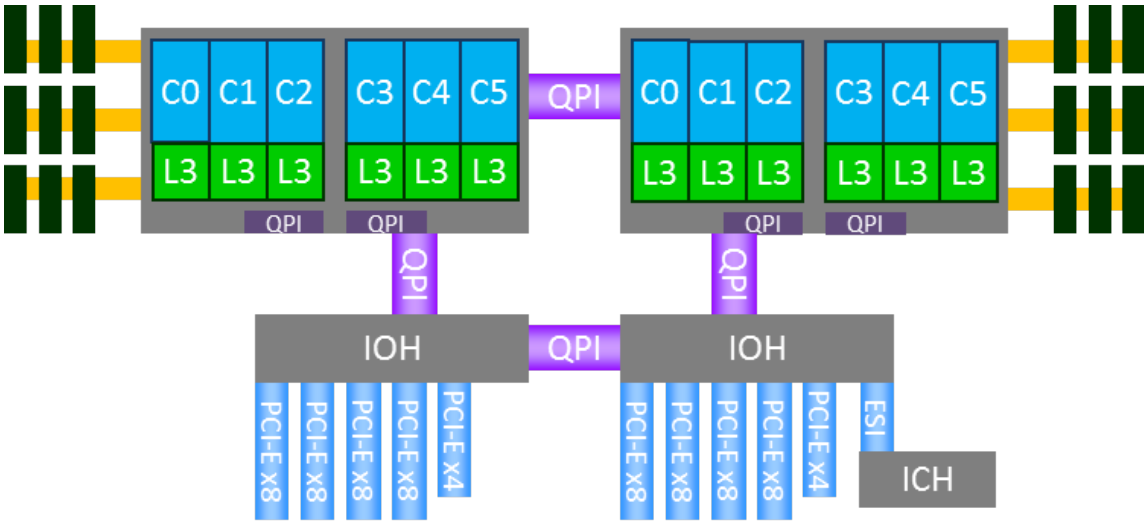


Figura 96: Arquitectura d'un node M610 amb les connexions QPI

A la Figura 97 mostrem un Nehalem de 4 nuclis amb la mateixa arquitectura que el Westmere, tot i que aquest últim disposa de 12MB de caché de L3. Veiem el càlcul d'ample de banda de 32GB/s pels tres canals. En un Westmere correspondria a 5.332GB/s / nucli.

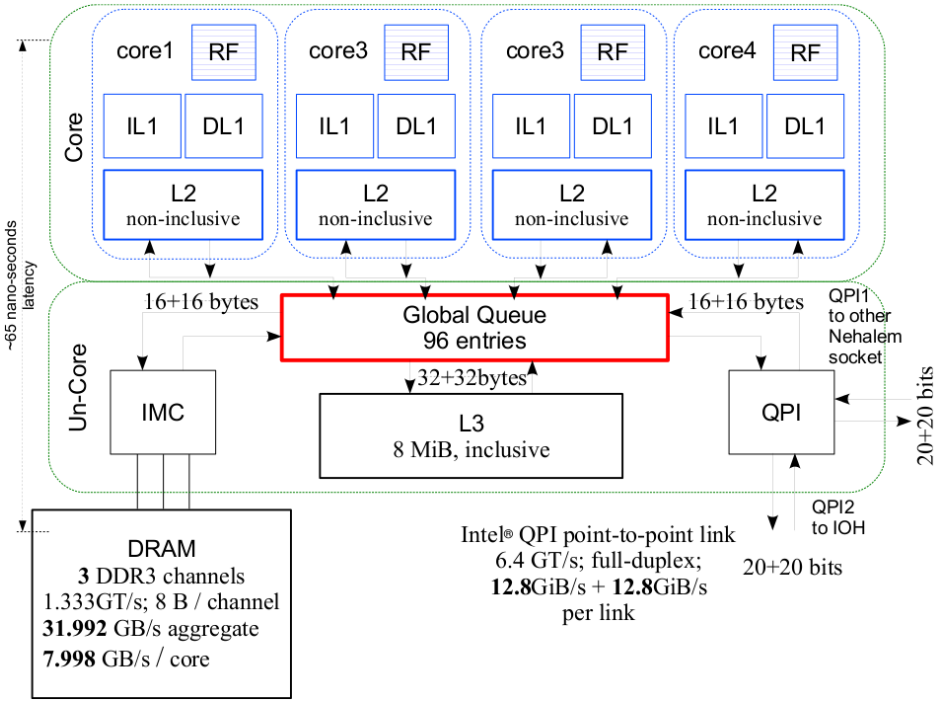


Figura 97: Arquitectura del controlador de memòria i tràfic de dades amb la cua global a un Nehalem de 4 nuclis

Al ser una arquitectura NUMA es requereix un protocol de coherència entre cachés de diferents nuclis i entre la memòria caché de nivell 3 i la principal. S'utilitza el protocol MESIF, una millora del protocol MESI típic en aquestes arquitectures. No explicarem aquí el protocol per no estendre massa el detall, tot i que posem un gràfic explicatiu del protocol bàsic MESI a la Figura 98.

A la figura veiem quina és la sortida de la comanda “lstopo” per els nodes M610 i que confirma l'arquitectura que estem detallant. Figura 99.

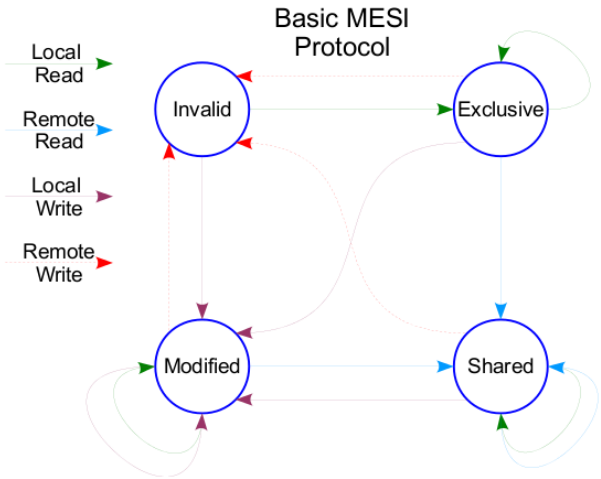


Figura 98: Protocol de cc-NUMA bàsic MESI

Gràcies a aquesta arquitectura podem mantenir la coherència de caché però obtenim una arquitectura NUMA que ens dona diferents latències d'accés a memòries locals i remotes.

A [41] s'analitza quin és l'accés a cada nivell de memòria local obtenint els següents resultats:

- L1 Caché (32KB) – Accés a un bloc de dades de 64KB amb 4 cicles de rellotge. Possibilitat d'obtenir-ne de nous cada 1 cicle (“throughput”).
- L2 Caché (256KB) – Accés a un bloc de dades de 64KB amb 10 cicles de rellotge.
- L3 Caché (12MB, compartida) – Accés a un bloc de dades ~35-40 cicles. Caché inclusiva, conté totes les dades de L1 i L2 per minimitzar l'impacte de l'”snoop” d'altres processadors cap a aquesta caché.

L'accés a memòria remota mitjançant els links QPI disminueix a 12.8GB/s, aproximadament un 40% de l'ample de banda teòric cap a la memòria RAM. La mateixa font realitza uns estudis per determinar quina és la diferència de rendiment entre l'accés local i remot.

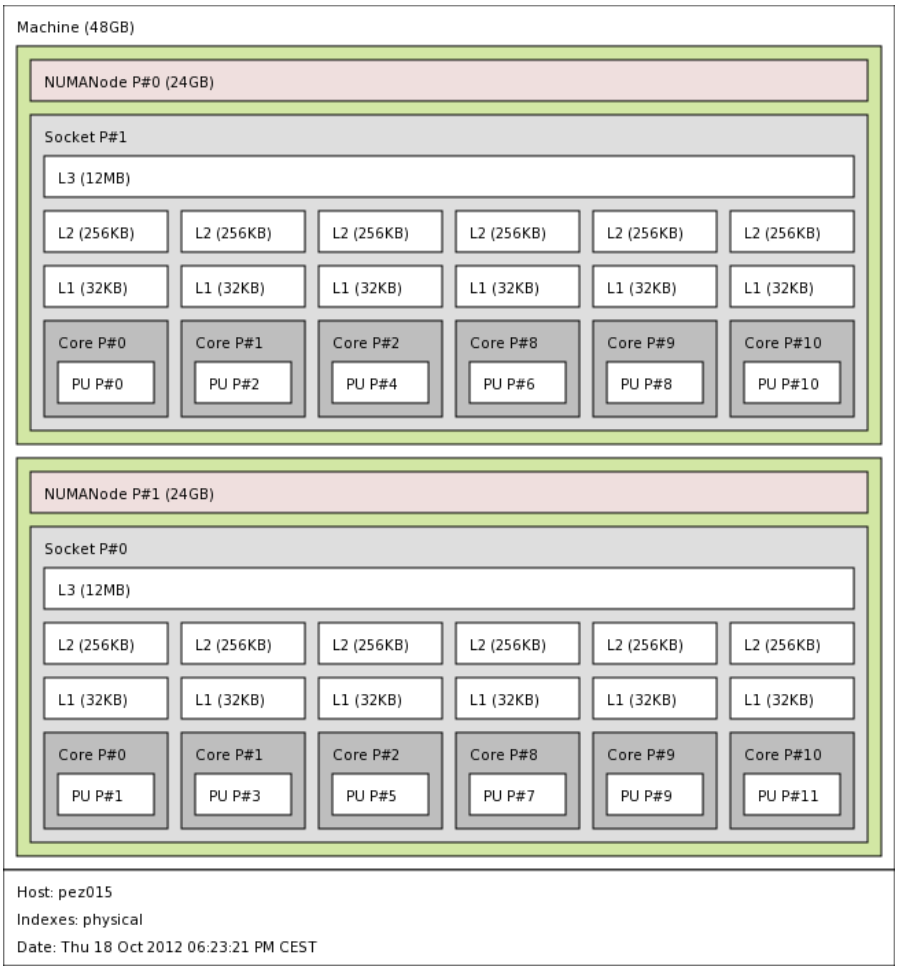


Figura 99: Sortida de lstopo per els nodes M610

Energia

En quant a l'eficiència energètica, segons Intel el processador Xeon E5645 treballa en el rang de 0.750-1.350V i pot consumir fins a 80W, dissipant una temperatura de 76.2ºC.

Una de les característiques més importants, a més d'integrar totes les dels seus predecessors, és la incorporació de la tecnologia Turbo Boost. Aquesta tecnologia, quan és activada des de la BIOS permet al processador desactivar els nuclis que no s'estan utilitzant i augmentar la freqüència dels que si que estan en ús per sobre del límit normal, fins a 2.67Ghz. Això augmenta el rendiment i a la vegada estalvia energia.

Un altra funcionalitat important és la capacitat de desactivar certes unitats funcionals que no es fan servir en els propis nuclis.

Amb aquestes millores i sumades a totes les de les anteriors arquitectures s'aconsegueix uns processadors eficients energèticament.

8.1.4 Comparació de les tres arquitectures

A la vista de l'estudi que hem realitzat podem fer una aproximació abstracte sobre les característiques principals de cada arquitectura i per quin tipus de codi ens poden ser més útils.

En primer lloc disposem dels nodes M600 amb l'Intel Xeon E5410 d'arquitectura Core2, la última que va disposar de Front Side Buffer. Aquests nodes de memòria UMA compartida posseeixen un bon rendiment per aplicacions que requereixen poc ample de banda de memòria i molt de càlcul de processador. Això és perquè encara que disposem de memòria FB-DIMM amb bon amples de banda, el FSB introdueix un coll d'ampolla molt important que en aplicacions de HPC serà decisiu en el bon rendiment de l'execució dels codis. Aquest coll d'ampolla és degut a que els programes que es solen realitzar a CIMNE requereixen molta memòria i no aprofiten molt bé les cachés de dades, creant per tant molt de tràfic de E/S.

En segon lloc disposem dels processadors Opteron 2356. El rendiment d'aquests en quant a processos que no requereixen ample de banda de memòria serà molt similar als del Xeon E5410, tot i que potser una mica inferior degut a que treballa a una freqüència més reduïda.

Per altra banda, al ser més eficaç en adquirir instruccions de 64 bits o SIMD fa al Barcelona un nucli ideal per tasques de HPC ja que aquestes instruccions són freqüents a l'àmbit de CIMNE.

Per altra banda podrem aprofitar molt bé l'HyperTransport en les aplicacions que maneguen gran quantitat de dades ja que tenim una connexió punt a punt entre processadors i memòria que elimina el coll d'ampolla del FSB.

En tercer lloc tenim els processadors Xeon E5645. Aquests processadors milloren molt la micro-arquitectura dels anteriors Core2 i proporcionen nous mecanismes que augmenten molt el rendiment dels càlculs i les instruccions per segon. Com a exemple podem fer els càlculs següents gràcies a que la implementació de les instruccions SIMD genera 4 operacions de coma flotant per cicle:

2.67GHz x 4FLOPs/Hz = 10,68 Giga FLOP / segon / nucli

10,68 * 6 nuclis = 64,08 Giga FLOP / segon / socket

64,08 * 2 sockets = 128,16 Giga FLOP / segon / node

Aquest gran rendiment ve a més millorat per l'avantatge de la tecnologia QPI. Si els AMD ja disposaven d'arquitectura NUMA, en els Nehalem s'implementa també aconseguint una millora substancial en els colls d'ampolla de l'antic FSB.

Per tant el millor processador del que disposem és clarament el E5645 i que ha de servir per realitzar càlculs que requereixin una alta potència.

Podríem determinar doncs que:

a) M600 (Xeon E5410) – Ideals per computacions que requereixen alta freqüència i poc ample de banda de memòria.

b) M605 (Opteron 2356) – Ideals per computacions que requereixen gran ample de banda

c) M610 (Xeon E5645) – Ideals per computacions que requereixen alt rendiment tant en instruccions per segon com amb ample de banda.

Per confirmar les deduccions que hem fet, he realitzat un estudi de l'ample de banda de la memòria als tres tipus de nodes.

8.1.4.1 Parell combinat de fils que fan lectura i escriptura concurrent

En aquest experiment un parell de fils llegiran i escriuran segments de memòria de mida entre 10KB i 200MB. La línia corresponent a CPU0-CPU0 mostra l'ample de banda observat quan el parell de fils treballen al mateix processador i no fan accés remot.

En el moment en que la capacitat de la caché de nivell 2 o 3 és esgotada, entra en joc el QPI i l'HT (en NUMA) i és qui imposa el límit d'ample de banda. En el cas del E5410 el mínim és marcat pel FSB. A les figures 100, 101 i 102 veiem aquests resultats.

Observem el resultats màxims i mínims en quant a l'ample de banda màxim (CPU0-CPU0):

	Caché L1	***Caché L2***	***Caché L3***	***Memòria P.***
M600 (Xeon E5410)	62.2 GB/s	20 GB/s	-	3.5 GB/s
M605 (Opter. 2356)	44 GB/s	17 GB/s	8.5 GB/s	4.4 GB/s
M610 (Xeon E5645)	70 GB/s	30 GB/s	20 GB/s	8 GB/s

També podem observar l'evolució entre comunicació de processadors més propers o més llunyans, determinat per la numeració CPUi.

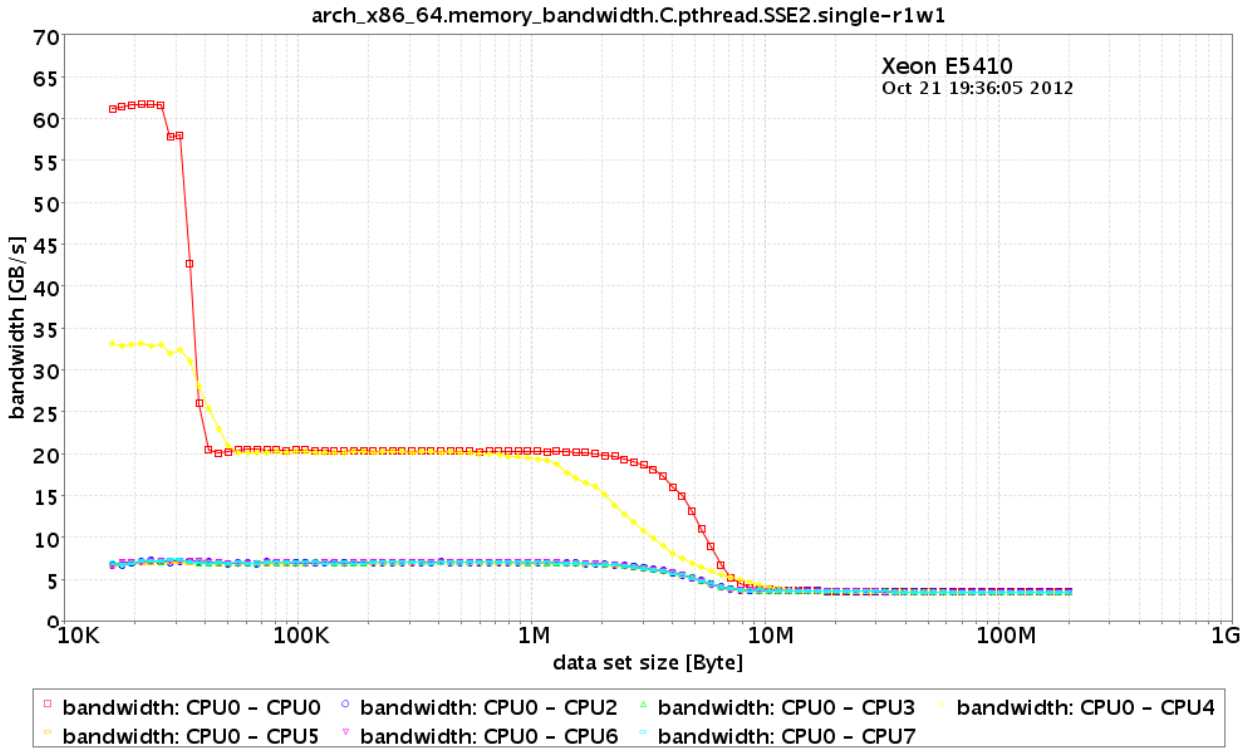


Figura 100: Parell de fils combinant lectura i escriptura - Node M600, Intel Xeon E5410

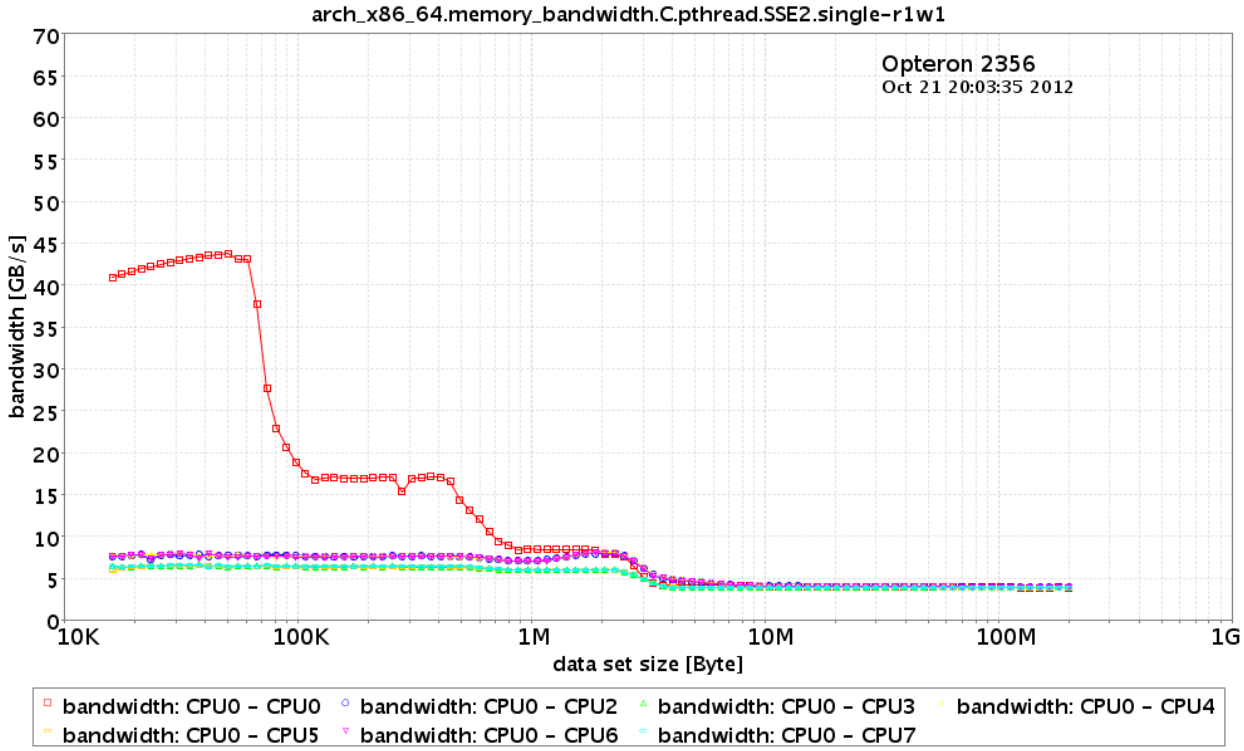


Figura 101: Parell de fils combinant lectura i escriptura - Node M605, AMD Opteron 2356

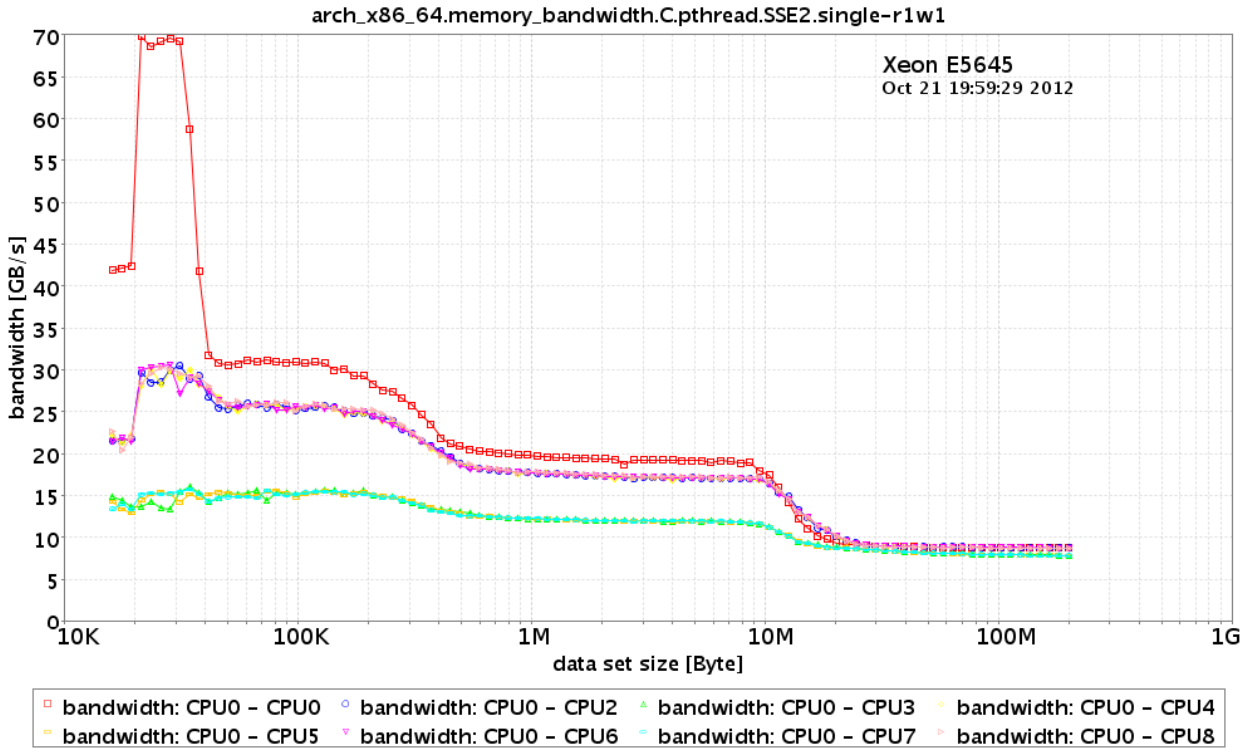


Figura 102: Parell de fils combinant lectura i escriptura - Node M610, Intel Xeon E5645

8.1.4.2 Múltiples parells de fils combinats de lectura i escriptura

Aquest experiment combina 8 parells com l'anterior. Es pot comprovar l'evolució més dràstica de la concurrència d'accés a memòria.

Veiem clarament que en els nodes M600 quan es sobrepassa el llindar de la caché el rendiment cau en picat, fet que també passa en les altres arquitectures NUMA però no amb tanta rapidesa.

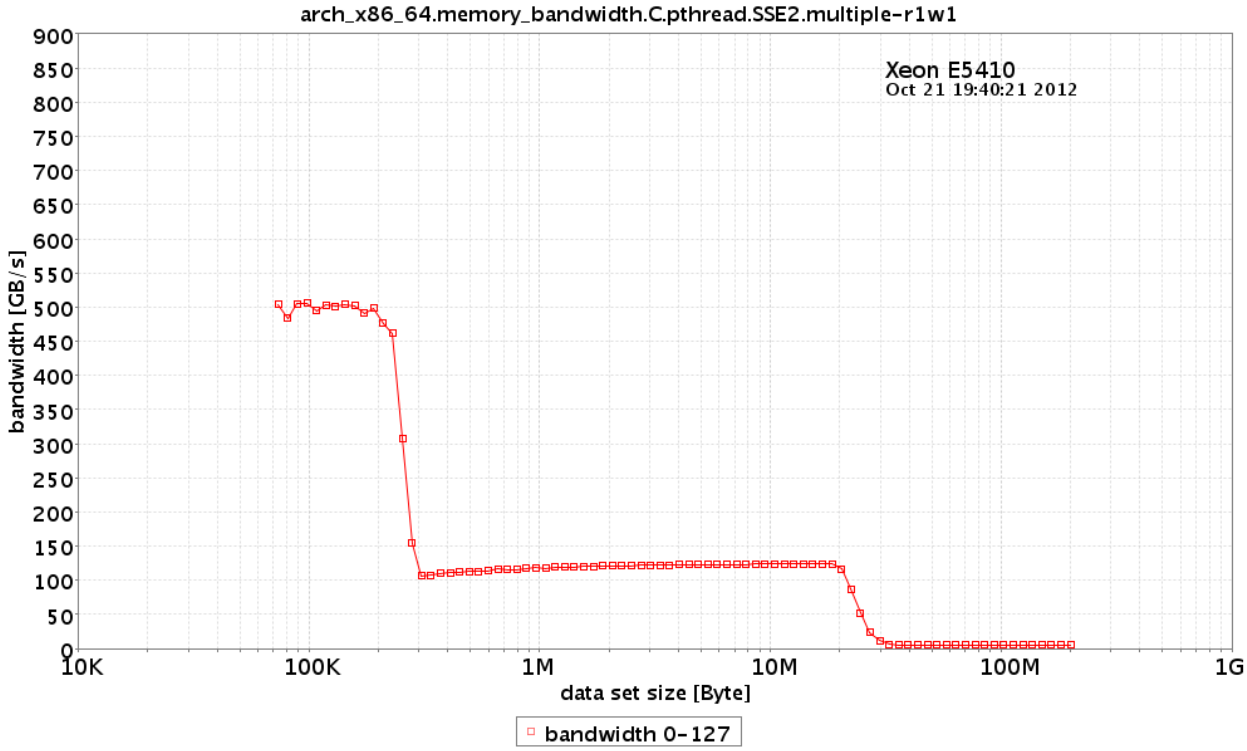


Figura 103: Parells múltiples de fils combinats de lectura i escriptura al M600, Xeon E5410

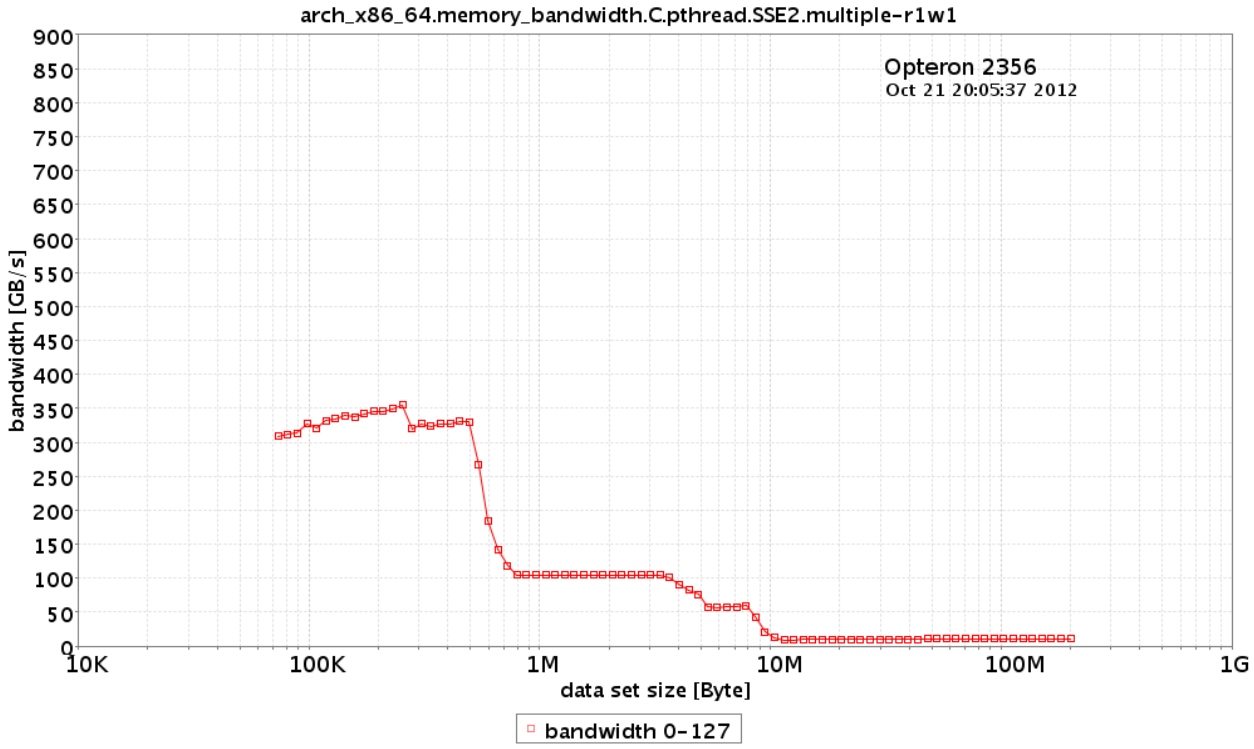


Figura 104: Parells múltiples de fils combinats de lectura i escriptura al M605, Opteron 2356

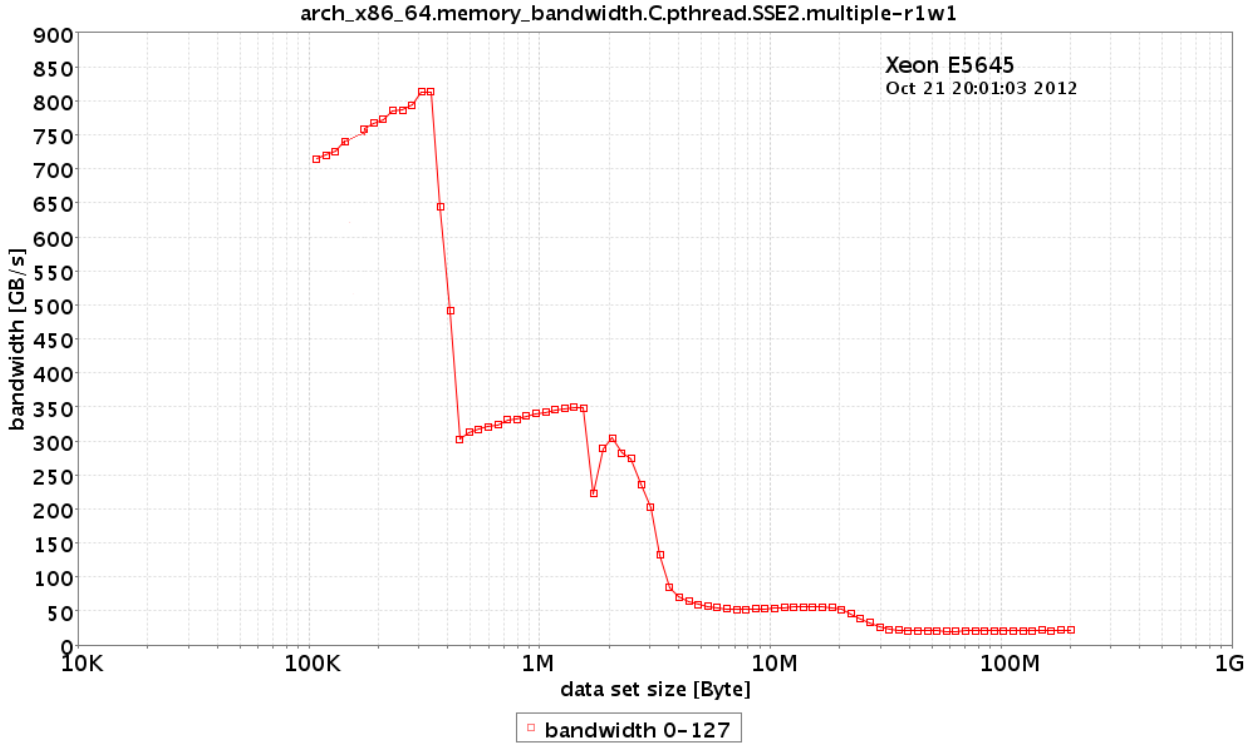


Figura 105: Parells múltiples de fils combinats de lectura i escriptura al M610, Xeon E5645

8.1.4.3 Producte escalar

Aquesta prova realitza el producte escalar sobre un vector. S'utilitzen diversos fils per fer les proves i es pretén demostrar el rendiment en quant a CPU i memòria utilitzats conjuntament, ja que ambdós factors hi influeixen.

A la Figura 106, Figura 107 i Figura 108 veiem els resultats de les proves que ens permeten determinar que el node més eficient en aquesta tasca és el Xeon E5645. Aquest manté constant el temps de càlcul en totes les mides de vectors provades quan s'envien a executar 8 fils paral·lels mentre que es mostra una escalabilitat amb una pendent molt suau en els altres casos.

En el cas de l'AMD 2356 ens ha sorprès el resultat de la bona escalabilitat del programa, mantenint en totes les proves una pendent constant i moderada. Tot i així els valors obtinguts mostren una mica menys de rendiment que el Westmere-EP.

En el cas del Xeon E5410 ens trobem amb un resultat molt confús. Veiem com existeixen molts outliers en els càlculs i que tot i mantenir una pendent també regulada i per sota del temps del Barcelona, ens fa pensar que els falls a caché i l'accés a memòria pel FSB ens fan tenir un rendiment general més baix. Per altra part cal destacar el problema que hem comentat constantment en aquest estudi i és que si saturem els dos FSB de cada socket (8 fils), obtenim un rendiment molt deteriorat.

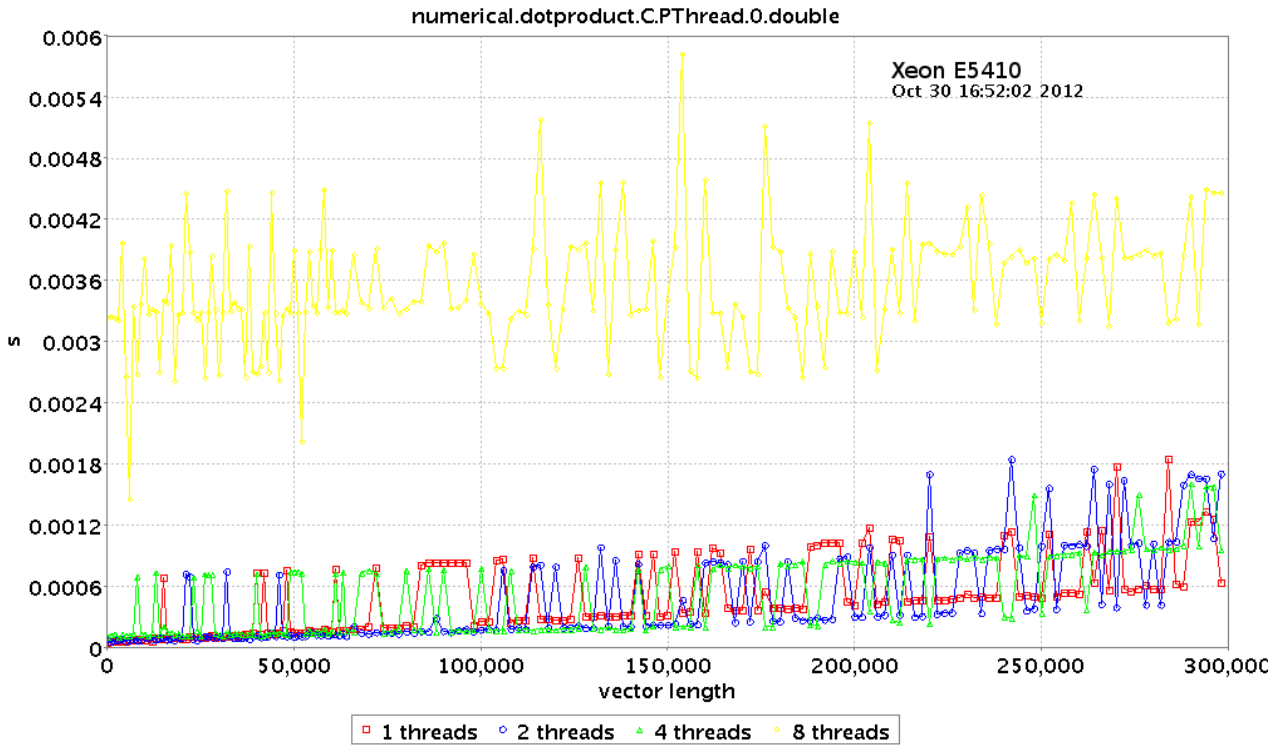


Figura 106: Producte escalar de vectors en un node M600, Xeon E5410

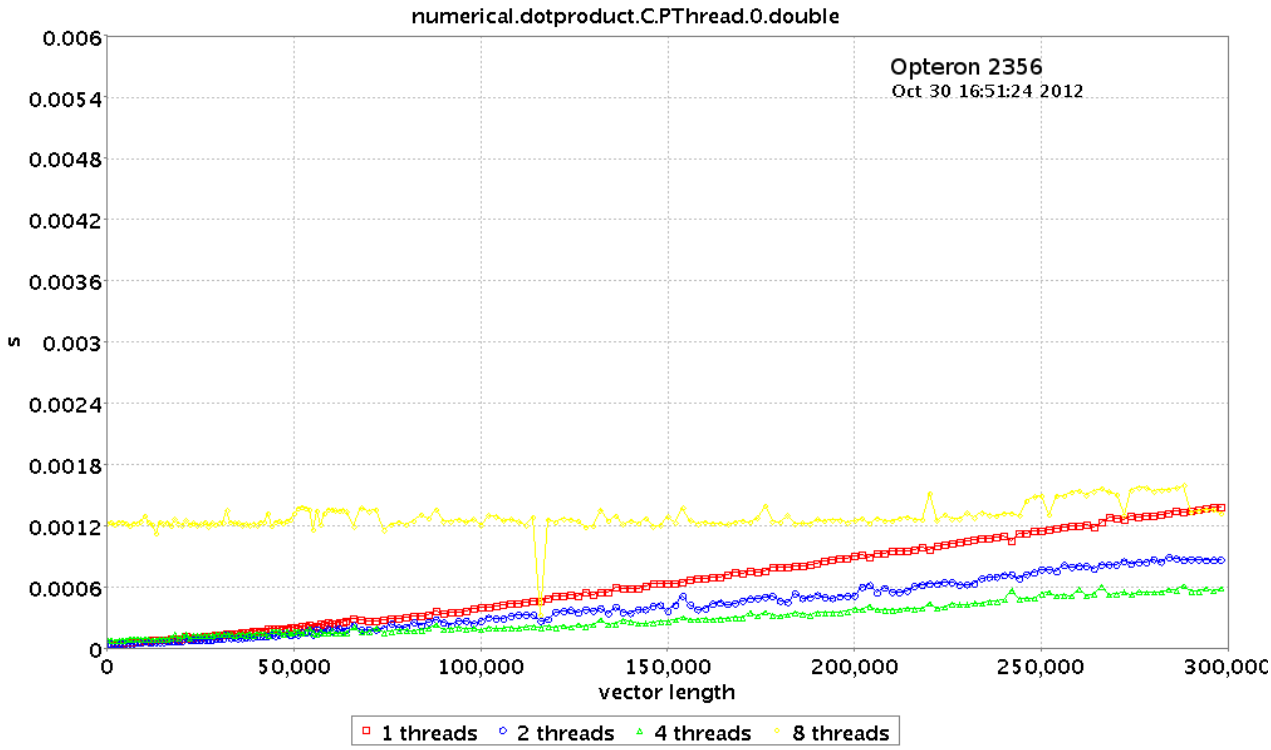


Figura 107: Producte escalar de vectors en un node M605, Opteron 2356

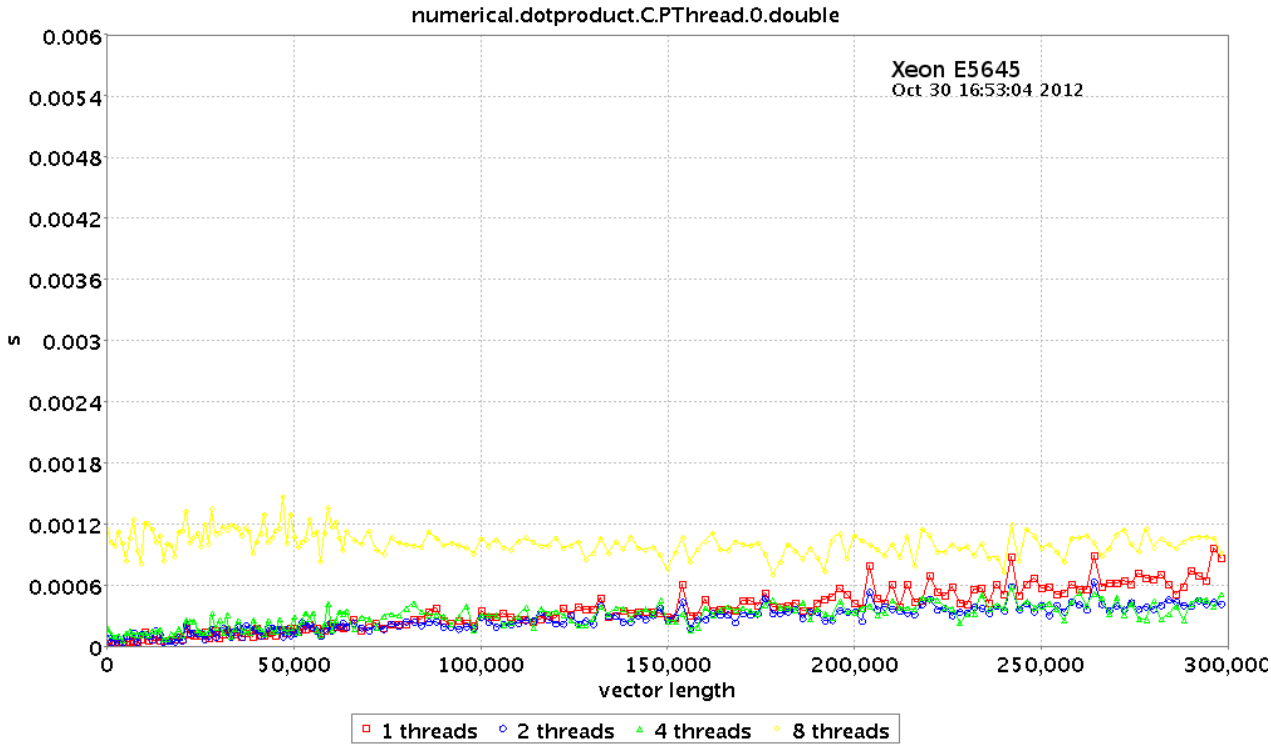


Figura 108: Producte escalar de vectors en un node M610, Xeon E5645

8.1.5 Implicacions al gestor de recursos

L'estudi ens mostra que tenim una mescla d'arquitectures NUMA i UMA que no ofereixen el mateix comportament en la majoria dels casos.

Sabem que el hardware s'encarrega de la comunicació entre processadors situats a diferents sockets i que crea una abstracció que allibera al sistema operatiu de gestionar la comunicació entre processadors. Això es realitza gràcies als protocols de cc-NUMA com el MESIF comentat abans i permet programar sense ser conscient de treballar amb NUMA.

Com a avantatge hi ha la portabilitat del codi i com a desavantatge la penalització per accés a memòria no local.

No obstant això, les aplicacions i el S.O. poden ser conscients que s'utilitza una arquitectura NUMA i poden gestionar la memòria per convenientment intentant aprofitar la localitat espacial.

Al nostre gestor de recursos haurem de tenir en compte quines són les “NUMA localities” i assignar a ser possible, processadors propers per una mateix tasca multi-fil.

Això significa que llançar un procés amb 6 fils a un Xeon E5645 implica que si no s'assignen a nuclis del mateix socket l'execució pot disminuir el rendiment si s'han de compartir dades. De la mateixa forma passarà amb els Opteron 2356.

Si més no, això no és cert per els nodes M600. Aquests disposen de dos sockets i cadascun un FSB, compartint la memòria. D'aquesta manera si llancem dos processos independents, cadascun amb dos fils s'assignen tots al mateix socket, tindrem un socket al 100% i el FSB creant un coll d'ampolla important. En canvi si pogués sim moure aquests dos processos un a cada socket disposaríem de dos FSB, un per cada procés.

Un cas diferent seria el de tenir una tasca multi-fil amb 4 processos on al assignar nuclis propers d'un mateix socket, tornaríem a crear un coll d'ampolla sobre aquell FSB. Per tant seria millor moure dos fils a un socket i dos a un altre.

L'hetereogeneitat del sistema fa que no puguem configurar SLURM de forma totalment correcta per tots els nodes, ja que hem de determinar si tenir una política de recursos agafant com a recurs el socket (CR_SOCKET a slurm.conf) o el nucli (CR_CORE).

Com a conclusió ja estem preparats per fer recomanacions als investigadors sobre quins nodes i per tant en quines particions els hi serà més beneficiós calcular. Cada cas serà especial i per tant haurà de ser analitzat, però gràcies a aquest estudi tenim una idea general que ens permet acomplir l'objectiu del treball de tenir capacitat de respondre els dubtes dels usuaris en aquestes questions.

8.2 Estudi de sostenibilitat

Aquest estudi de sostenibilitat pretén mostrar quin és el consum actual del clúster i quines mesures hem pres per millorar-lo. Al final farem un comentari sobre el reciclatge o reutilització dels antics XFire i Vega.

8.2.1 Mètriques de consum

Les mesures de consum que prendrem seran els W (J/s) totals consumits pels diferents elements.

Cada KWh significarà una despesa en € per CIMNE i una emissió de gasos CO^2 a l'atmosfera, per tant el que cercarem és reduir al màxim els KW totals.

Ens centrarem en diversos nivells d'abstracció per analitzar quin impacte té cada element. Començarem per el nivell més baix que són els processadors dels nodes i acabarem per veure en general que consumeix el servei de càlcul.

Haurem de mesurar, per tant, els següents elements:

- Consum per processador, memòria i discs durs.
  - En aquest cas la memòria i els discs durs no poden ser mesurats de forma dinàmica directament, sinó que només disposem del TDP (màxim consum).
- Consum d'un node
- Chassis complet
- Cabina de discs
- Switchs
- Servei de càlcul

A data de 20/10/2012, segons dades d'Iberdrola [2] i referint-nos a la tarifa 2.1A, el preu en €/kWh és de 0,17469.

8.2.1.1 Obtenció de dades

Per obtenir el consum ho farem de diverses formes segons el nivell que vulguem analitzar.

En primer lloc disposem de la funcionalitat IPMI als processadors i de les especificacions tècniques d'aquests a més de també les dels mòduls de RAM i discs durs.

És possible que actualment sentim a parlar de la tecnologia RAPL, “Running Average Power Limit”. Aquesta tecnologia permet mesurar molt més acuradament el consum i ha estat introduïda en la micro-arquitectura SandyBridge, per tant nosaltres no disposem de la capacitat.

En segon lloc podrem obtenir un consum general del node a la interfície iDRAC del CMC, on a més obtindrem el consum global del chassis.

La cabina de discs haurà de ser monitoritzada amb el Dell MD Storage Manager i també mirant el consum de les fonts d'energia.

Per el servei de càlcul farem la suma dels elements anteriors però també accedirem al recurs de la UPC Sirena que monitoritza i fa un històric de consum en els edificis del Campus Nord, [66].

Diferenciarem també entre KW i KWh, essent el primer una unitat de potència en Joules / segon i la segona, una unitat d'energia consumida en una hora. El primer s'utilitza com a consum mesurat puntualment, i el segon serveix per calcular les tarifes.

8.2.2 Mesures de consum energètic

Hem realitzat una sèrie de mesures que detallem a continuació.

Consums màxims teòrics

Els següents consums han estat obtinguts de les especificacions dels components.

- Node M600

Processador Intel Xeon E5410	80W
Mòdul 2GB DDR2 FB-DIMM 667 (NT2GT72U4NB1BN-3C, 1.8V±0.1V)	10W

- Node M605

Processador AMD Opteron 2356	75W
Mòdul 4GB DDR2 667MHz ECC (M3 93T5160CZA-CE6, 1.8V±0.1V)	14W

- Node M610

Processador Intel Xeon E5645	80W
Mòdul 8GB DDR3 1333MHz ECC (HMT31GR7BFR4C-H9, 1.5V±0.075V)	10W

- Chassis + Nodes 7.798W

Consum màxim observat entre 05/2008 i 10/2012

Aquestes dades són un resum de la informació del CMC on hi podem trobar estadístiques de consum de tots els nodes i el chassis.

Nodes M600	344 W	08 Gener 2012, 07:36h
Nodes M605	360 W	01 Març 2012, 03:57h
Nodes M610	341 W	29 Gener 2012, 04:29h
Chassis + Nodes	3.416 W	10 Octubre 2012, 18:17h

Hem pogut comprovar efectivament com en aquestes dates el monitor de Ganglia mostra quins són els nodes amb una carrega intensa a les dates en que més energia consumeixen. En el pitjor cas dels nodes M605 es correspon al node Pez014.

Consum mitjà dels nodes observat

Per realitzar aquesta mesura hem esperat a un moment on el clúster estigués amb una càrrega de treball mitjana i hem fet la mitja per cada tipus de node. Els resultats han estat els següents:

Nodes M600	232 W
Nodes M605	135 W
Nodes M610	161 W
Chassis + Nodes	3.072 W

Consum mínim observat entre 05/2008 i 10/2012

Obtingudes de l'històric del CMC ens permet determinar quina és la menor quantitat d'energia que ha utilitzat mai el clúster en un moment puntual.

Veiem que aquest mínim és molt proper al pic d'energia sofert el 10 de Març i es correspon a una baixada del nombre de treballs al clúster i per tant de la càrrega general. Observant l'eina sacct i el monitor Ganglia hem confirmat com en aquell moment tots els processos que hi havia corrent, excepte els del nodes 13 i 11, van acabar i el clúster va quedar alliberat, Figura 109.

Hem pogut veure també quin consum mínim tenen els nodes de forma individual:

Nodes M600	168 W
Nodes M605	124 W
Nodes M610	80 W
Chassis + Nodes:	2.556W , 10 Octubre 2012, 18:43h

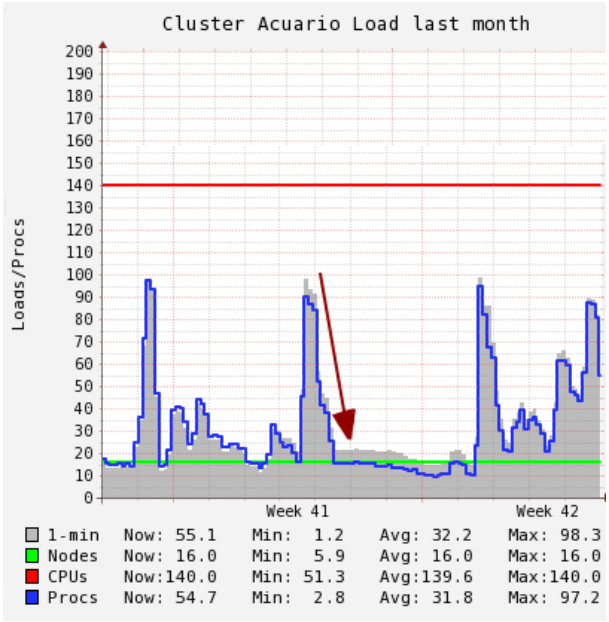


Figura 109: Baixada del nombre de processos el 10 d'Octubre de 2012

Consum mesurat al SAI 16/05/2011 i 19/10/2012

El SAI ens proporciona una mesura de l'energia consumida en cada moment per les fases generals que arriben als racks.

Agafant la mesura del rack complet del clúster i comparant-la amb el consum llegit pel chassis + nodes des de l'eina CMC, podem determinar quin és el consum dels elements restants, és a dir, switch i cabina de discs.

En un moment determinat a data 20/10/2012 a les 19h hem obtingut que a la fase L3 s'hi proporcionen 230V i 14,3A, donant el consum de:

Consum a la fase del R3 = 230V * 14,3A = 3.289W

Aquesta càrrega de 3.289W correspon a un 20% del total del CPD de CIMNE.

En aquest mateix instant el CMC mesurava un consum de chassis + nodes de 3.096W amb pics de 3.416W. Agafant la xifra de 3.096W podem determinar el que el SAN i el Switch consumien en aquell moment:

Consum de SAN i Switch = 3.289W – 3.096W ~= **193 W**

Consum acumulat mesurat entre 16/05/2011 i 19/10/2012

El consum mesurat del chassis + nodes en aquest període ha estat de: **39.225,8 kWh**

Aquest consum ens dona una fita important al poder determinar quin és el consum mitjà en un marge de temps raonable. Per exemple, si volem trobar el cost diari, mensual, anual, etc. realitzem el següent càlcul:

39.225,8kWh / 522 dies = 75,145 kWh/dia

Preu diari	=	75,145 * 0,17469	=	**13,12 €/dia**
Preu mensual	=	13,12 * 30 * 0,17469	=	**393,81 €/mes**
Preu anual	=	13,12 * 365	=	**4.788,80 €/any**

A tot això no hem d'oblidar de sumar-li el consum de la cabina de discs i dels switchs.

Modificant doncs el càlcul anterior i tenint en compte els ~200W de l'apartat anterior:

200 W * 24h = 4,8 kWh/dia

Preu diari	=	4,8 * 0,17469	=	**0,84 €/dia**
Preu mensual	=	0,839 * 30 * 0,17469	=	**4,39 €/mes**
Preu anual	=	0,839 * 365	=	**306,06 €/any**

Consum energètic anual observat

Del total de kW diaris consumits calculats obtenim el consum energètic del servei de càlcul anual:

(75,145kWh/dia + 4,8kWh/dia) * 365 dies = 29.180 kWh / any

29.180 kWh/any

Taula 12: Despesa energètica anual del servei de càlcul

Despesa econòmica energètica anual

Per tant el consum total del servei de càlcul durant un any amb la configuració actual és:

4.788,80€ + 306,06€ = 5.094,86 €/any

5.094,86 € / any

Taula 13: Despesa econòmica anual del servei de càlcul

CO^2 emès

De formes similars podem fer el mateix càlcul per determinar la quantitat de CO^2 emes a l'atmosfera.

En total haviem determinat que el servei de càlcul consumia:

75,145 kWh/dia + 4,8kWh/dia ~= **80 kWh/dia**

Segons [67], 1 kWh hora equival a 0,5246 kg CO^2 d'energia provinent de la xarxa elèctrica.

Per tant podem concloure que el servei de càlcul emet:

80 kWh/dia * 0,5246	=	42 kg CO^2 / dia
42kg CO^2/dia *365d	=	15,30 TN CO^2 / any

42 kg CO^2 / dia **15,30 TN CO^2 / any**

Taula 14: Emissions de CO^2 equivalent a l'atmosfera

Càlcul estimat per treball

Havent obtingut la mesura de consum del servei de càlcul per any podem fer estimacions respecte el preu del temps de càlcul.

Hem d'agafar el temps de càlcul efectiu durant un any i dividir-lo amb aquest consum, d'aquesta manera obtindrem el preu per temps que ha costat el càlcul d'aquests treballs.

Fent servir l'eina sr*eport* de la següent manera obtenim el temps total d'utilització del clúster per el període de 20/10/2011 a 20/10/2012.

```
[fmoll@acuario ~]$ sreport cluster Utilization Start=102011 End=102012
--------------------------------------------------------------------------------
Cluster Utilization 2011-10-20T00:00:00 - 2012-10-19T23:59:59 (31622400*cpus secs)
Time reported in CPU Minutes
--------------------------------------------------------------------------------
  Cluster    Allocated       Down  PLND Down         Idle  Reserved     Reported
--------- ------------ ---------- ---------- ------------ --------- ------------
  acuario     28847054    3808446     117991     21315988   1102513     55191992
```

En total surten a l'estadística 55.191.992 minuts. Aquests minuts no són exactes degut a talls de llum, a parades del servei en algunes ocasions i en que en aquesta franja de temps el clúster ha sofert modificacions en el nombre de CPUs, particions, etc. No obstant s'aproxima al que volem calcular.

El temps total utilitzat en càlculs són 28.847.054 minuts entre tots els recursos disponibles al sistema, que en total disposa de 12*8 + 3*12 = 132 nuclis i per tant correspondrien a 3.642 hores de càlcul per cada nucli.

Això fa que tinguem el clúster al 100% d'utilització de tots els recursos disponibles durant un total de 3.642h / 24h = ~152 dies.

Partint dels consum següent:

Consum servei de càlcul = 5.094,86 € / any

Si de 365 dies hem tingut ocupat al 100% (tots els nuclis) el clúster 152 dies (un 41% del temps), el preu per dia de càlcul ens ha sortit a:

5.094,86€ / 152 dies = 33,51€ / dia

Què convertit a hores:

33,51 € / 24h = 1,40 €/h

Obtenim finalment l'amortització de l'equipament:

1,40 €/h / 132 nuclis = 0,010606 €/h / nucli

0,010606 € / h / nucli

Taula 15: Preu per hora de càlcul amb un procés sèrie al servei de càlcul CIMNE

Aquesta xifra ens dona un indicador de l'amortització de l'equipament.

Com més baix sigui aquest nombre, encara que s'incrementés una mica el consum global, el rendiment preu/hora i la relació ús/emissions CO^2 seria millor.

El consum ideal en aquest cas s'hauria de calcular suposant que hem tingut els nuclis al 100% durant 1 any sencer. D'aquesta manera introduïm un factor d'increment de consum energètic del 40% per fer més realista l'estudi.

(75,145kWh/dia + 4,8kWh/dia) * 365 dies * 40% = 40.852 kWh / any

40.852 kWh / any * 0,17469 €/kWh = 7136,44 € / any

7136,44€ / any / 132 nuclis = 54,06 € / any / nucli

Convertint finalment el preu per any a preu per hores, aquest exemple d'amortització al 100% durant un any sortiria a:

46,34 € / any / nucli / 365 dies / 24 h = 0,0062 €/h

Aquest és per tant un valor d'amortització ideal d'amortització a aconseguir, tenint el clúster el 100% durant un any i consumint un 40% mes del que hem mesurat anteriorment.

0,0062 € / h / nucli

Taula 16: Amortització ideal. Preu de càlcul per hora per nucli suposant una càrrega del 100% durant 1 any i suposant un increment en el consum d'un 40%.

8.2.3 Millores aplicables al servei

Processadors

Després de realitzar l'anàlisi en profunditat de l'arquitectura del hardware hem pogut observar com el processador és responsable d'una bona part de l'energia consumida. És interessant llavors centrar-se en aquest aspecte i intentar activar les funcionalitats que disposa cada processador.

Una de les primeres funcionalitats que podem activar a les BIOS és la funcionalitat de l'escalat de la freqüència dinàmica, el Demand Based Switching ja comentat. Aquesta tecnologia juntament amb els kernels actuals de SL 6.1 modificarà les freqüències quan els nodes estiguin inactius.

La segona funcionalitat, més delicada, és la del mode Turbo en els M610. Aquest mode permet desactivar fins a 0V els nuclis que no s'estan fent servir i augmentar la freqüència del que si que és en ús. Aquesta mesura però, pot tenir influències en l'escalat d'alguns programes i s'ha d'estudiar l'impacte que pot tenir. No obstant hem activat l'opció i hem pogut comprovar com els nodes M610 arriben a una potència mínima de 80W per contra dels altres que no baixen dels 124W.

Memòria RAM

Les memòries actuals no són de baix consum. Amb memòries DDR3 de baix voltatge (només nodes M610) que funcionin a 1.35V enlloc dels actuals 1.8V aconseguiriem aproximadament entre un 10 i un 15% menys de consum, de 10W a 8.5W. Tenint en compte que disposem de 6 mòduls en els M610 el consum de la memòria podria veure's reduït en un cas de càrrega màxima dels mòduls, de 60W a 49W per node.

Discs durs

Els discs durs dels nodes també consumeixen. És per això que és desitjable disposar de la funcionalitat del kernel de Linux d'enviar l'ordre d'apagat en els discs que no s'estiguin utilitzant. De fet l'ús del disc local és poc si es treballa per xarxa i l'impacte d'aquesta mesura, implementada per defecte, no és gran.

Aprofitament dels nodes

Si bé aquesta mesura no és directament una forma de reduir el consum, s'ha de tenir en compte per l'amortització dels equipaments i per el rendiment de kg equivalent de CO^2 emès / hora de càlcul.

Com hem pogut veure a l'apartat “Càlcul estimat per treball“, augmentar la utilització del servei millora la seva eficiència energètica i n'abarateix la despesa monetaria.

8.2.4 Estadístiques d'UPC

La pàgina web de la UPC ens posa a disposició una eina per comprovar quina és la despesa energètica realitzada per cada edifici al llarg dels anys [66].

Obtenint les seves gràfiques podem veure com el consum del C1 el 2012 és de 360.092kWh, essent per tant el clúster el responsable del 21.980kWh/360.092kWh = 6,10% del consum.

Si mirem l'estadística del 2010 i 2011 veiem que el consum és de l'ordre de 575.000kWh, i per tant veiem com hi ha hagut una baixada de 214.908 kWh. Evidentment d'aquesta baixada no n'ha estat responsable el clúster però si que ha coincidit en el moment en que es portava a terme aquest projecte. El responsable d'aquesta baixada són les retallades que ha sofert la UPC en l'últim any, on s'han parat aires condicionats, llum tot el mes d'agost i altres reduccions.

El que si que ens permetrà fer una comparació és l'edifici C2. Aquest edifici disposa d'un clúster semblant al de les nostres característiques i podem comparar quina baixada de consum ha tingut respecte a nosaltres, ja que se li han aplicat les mateixes polítiques d'estalvi.

Edifici C1 (kWh/any)	**Edifici C2 (kWh/any)**
2010 - 579.391	2010 - 259.936
2011 - 573.910	2011 - 277.687
2012 - 360.092	2012 - 224.412

La diferència és d'una reducció del consum ha estat d'un 37,25% per l'edifici C1 i un 20% pel C2. Aquest 17,25% de diferència pot incloure alguna millora com la virtualització de molts de servidors realitzats pel departament de Sistemes, a més de un petit percentatge del clúster i finalment una diferència en l'ús del laboratori de RMEE.

Mostrem a la Figura 110 una gràfica de l'any 2011, moment en que es va implementar el nou servei. Ens podem fixar especialment en els mesos d'octubre, novembre i desembre, on el clúster va ser finalment migrat del tot. Notem la diferència en els tres mesos de 40.355kWh, 35.245kWh i 33.639 kWh respectivament.

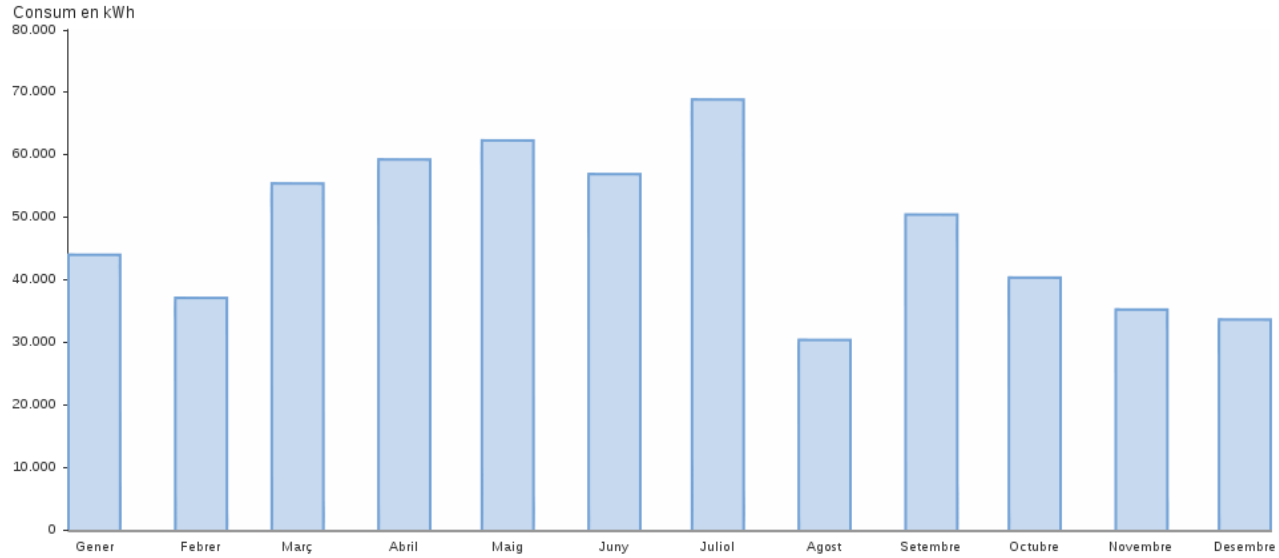


Figura 110: Consum de l'edifici C1 en kWh a l'any 2011

8.2.5 Reciclatge de XFire i Vega

Durant les proves realitzades hem pogut mesurar el consum d'aquests dos servidors basant-nos amb els registres del SAI.

En condicions de càrrega normal al rack 2, sense engegar XFire ni Vega i només amb els 3 servidors restants, hem mesurat un consum energètic de 13,7A i 230V, donant un total de 3.151W.

Engegant XFire i esperant a que el sistema s'estabilitzés (sense executar cap procés de càlcul) hem pogut comprovar com l'entrada a les fases del rack 2 passava de 13,7A a 16,4A.

D'aquesta manera podem afirmar que XFire consumeix, en un estat de inactivitat:

Consum de XFire inactiu: 2,7A * 230V = **621W**

Per altra banda, realitzant el mateix procediment amb Vega hem obtingut el consum de:

Consum de Vega inactiu: 1,9A * 230V = **437W**

L'eliminació d'aquests dos servidors ha suposat la baixada del consum energètic en el rack R2 d'almenys 1058W.

Consum anual estimat

El consum elèctric anual d'aquests servidors en estat inactiu era de:

1058 W * 24h * 365 dies = 9.268,08 kWh / any

Despesa energètica anual

La despesa que suposava tenir aquests dos servidors engegats en estat inactiu suposava un total de:

0,17469 €/kWh * 9.268,080 kWh / any = 1.620€

Estalvi aconseguit

Si tenim en compte que només feien servir aquests servidors un màxim de 4 usuaris:

1.620€ / 4 = **405 € / any / usuari**

Aquest preu és molt elevat comparant-lo amb l'actual consum que pot efectuar un procés sèrie durant un any al clúster:

0,010606 €/h / nucli * 365 dies * 24 h = **93,9 € / any**

Ja que a més, els 405€ que hem calculat es basen en un estat d'inactivitat total dels servidors.

Si suposem que aquests servidors per el preu de 1.620€ estaven ocupats al 100% (8 + 4 nuclis), podem veure que si els migrem al clúster obtindrem un benefici de:

1620€ - 12*93,9€ = 493,2 € / any

493,2 € / any

Taula 17: Benefici obtingut per apagar els dos servidors XFire i Vega i migrar els usuaris al clúster

Aprofitament dels servidors

L'aprofitament d'aquests servidors ha quedat per determinar i el departament està avaluant diverses possibilitats.

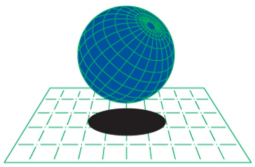
a) Una és la d'utilitzar els servidors com a servidors de proves on poder experimentar amb el hardware i software, o amb altres programes.

b) Una altra és la d'afegir aquest servidors a les cues, opció que sembla poc factible degut al seu elevat consum i poc rendiment.

c) Vendre els servidors a terceres empreses que necessitin aquests models, recuperant així una petita part de la inversió.

d) Finalment queda la opció de portar els servidors al punt verd i fer-ne un reciclatge correcte.

Les opcions més apropiades són la c) o la d), ja que es pot treure poc aprofitament i més tenint el clúster no utilitzat al 100%.



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Capítol 9
Conclusions

9.1 Entrada en millora continua

Seguint amb l'esquema ITILv3 que comentàvem al principi d'aquest projecte i una vegada realitzar totes les etapes prèvies és moment d'entrar en la millora continua del servei.

Això té una sèrie d'implicacions naturals que comentem a continuació.

Aquest projecte tenia un objectius marcats des del principi, no obstant això és un projecte sense fi perquè sempre sorgeixen noves necessitats, nous problemes, i s'han de corregir errors ja que no hi ha cap sistema perfecte.

La millora continua del servei passa per monitoritzar tot el que succeeix al clúster i adonar-se de quines noves necessitats apareixen i quins nous objectius a acomplir. D'aquesta manera, durant la realització del treball he pogut comprovar com el que anava fent a vegades quedava obsolet o s'havia de retocar. Doncs bé, és el moment de fer això i planejar els objectius de futur i les accions que donaran continuïtat a aquest projecte.

La realització del projecte ha estat satisfactòria i m'ha obert moltes portes que em permetran endinsar-me cada cop més a l'extens món del HPC. Com a exemples i en vista de l'èxit que ha tingut aquest projecte ja s'està parlant d'adquirir un nou clúster de computació que hauria de ser integrat amb l'actual sistema de cues. Per altra banda hi ha rumors de que un nou node d'accés pot ser adquirit i que alliberaria al node 0, Acuario i permetria integrar-lo al sistema de càlcul.

També es parla d'ampliar la cabina de discs o de millorar el sistema de fitxers.

Totes aquestes coses entren en la millora continua que forma un procés circular i sense fi.

El meu objectiu a partir d'ara serà mantenir i aportar valor a aquest nou projecte finalitzat.

9.2 Objectius realitzats

Al capítol 1, Introducció, vam xerrar dels Objectius del projecte. En el capítol 2, Anàlisis de situació actual vam analitzar quins serien els requisits funcionals i no funcionals a implementar.

Doncs bé, podem dir que s'han aconseguit el 100% dels objectius, obtenint un servei de càlcul complet, funcional, eficient en tot el que es pot, i havent adquirit un coneixement que ja no serà excusa per no poder contestar als usuaris en relació als problemes de càlcul que puguin tenir.

En quant als requisits no funcionals també hem intentat complir-los tots i sota el meu punt de vista crec que s'han aconseguit al 100%.

És en els requisits funcionals la part que han quedat pendents algunes tasques. No obstant això aquests han estat problemes no directament relacionats amb el treball sinó externs i per configuracions d'altres serveis.

Els que no s'han pogut acomplir són:

1. Implementar un sistema de control de comptes d'usuari, un sistema de seguretat, crear polítiques d'ús i restringir permisos dels usuaris. Integrar-ho amb l'LDAP del CIMNE.
2. Proporcionar un sistema de monitoreig de l'estat del clúster. Configurar els components bàsics per la integració en el sistema de monitorització del departament de Sistemes basat en Nagios.

Tot i això, en el primer cas en que no hem pogut integrar el sistema amb LDAP, si que hem proporcionat un sistema alternatiu, el NIS, que ens permet controlar els usuaris de forma local.

En el cas de Nagios s'han proporcionat suficients eines per portar un control acurat del que passa al servei de càlcul, si més no en el futur sempre es podrà integrar amb el que implementi CIMNE.

9.3 Planificació temporal inicial vs planificació final

La planificació temporal en quant al nombre d'hores de feina ha estat més o menys acurada.

El que no ha coincidit correctament han estat les etapes que vam planificar ja que hi ha hagut molts solapaments entre tasques.

Per exemple, al realitzar la instal·lació anaven sorgint dubtes que portaven a investigar noves tecnologies i a mirar el perquè de segons quines coses.

També hi ha hagut problemes amb les planificacions del desplegament que ja hem comentat anteriorment, com que els usuaris ens impedien realitzar la conquesta de nodes en els dies que havíem senyalat.

Hem modificat també algunes etapes inicials, com les de Validació i verificació que han estat implícites dins els capítols d'Implementació i desplegament.

En quant a les hores dedicades a la memòria del projecte i a la documentació en general, ens hem estès massa i hem sobrepassat el límit de 176 hores que havíem planificat, amb un desviament d'unes 40 hores.

Per altra banda les hores dedicades a etapes com les de planificació o estudis han estat molt encertades.

Comparant-ho amb la taula que vam fer a l'inici d'aquest projecte obtenim el següent:

Planificació prevista:

- Planificació:
  72h (18 dies)
- Investigació:
  185h (47 dies)
- Implementació:
  208h (53 dies)
  - Verificació i validació:
    32h (8 dies)
- Desplegament:
  23h (6 dies)
  - Avaluació, estudis i proves:
    120h (30 dies)
- Documentació: 176h (44 dies)

Planificació realitzada

- Planificació:
  72h (18 dies)
- Investigació:
  185h (47 dies)
- Implementació:
  228h (57 dies)
  - **Verificació i validació:**
    20h (5 dies)
- Desplegament:
  132h (33 dies)
  - **Verificació i validació:**
    12h (3 dies)
  - Avaluació, estudis i proves:
    120h (30 dies)
- **Documentació:**
  216h (54 dies)

En total podem dir que de 817 hores planificades haurem realitzat realment un total d'unes 860 hores, un 5% més del temps planificat. No obstant això, les entregues han estat realitzades dins el temps previst, entre Abril i Octubre de 2011.

9.4 Cost final del projecte

Tenint en compte les hores de més realitzades i que no s'han utilitzat equipaments més enllà del que estava planificat, hem calculat que:

1. Les hores pagades la treballador han passat de 11.846,5€ a 12.470€.
2. L'ús de l'ordinador personal ha passat de 20€ a 22€

Per tant el preu total s'ha incrementat en 625,5€, quedant un total de:

Cost total final del projecte
12.533,31 €

Preu encara molt assequible si el comparem amb els pressupostos inicials que van presentar altres empreses externes a CIMNE.

9.5 Treball futur

Arribats a aquest punt del projecte i després de tota l'experiència recopilada, han sorgit diverses idees per millorar l'actual servei de càlcul, les quals exposem a continuació:

– Plugin de SLURM “sbank”

  Es tracta d'implementar aquest plugin consistent en un conjunt d'scripts que fan servir les eines de sacctmgr, sshare, sinfo i sacct i que analitza les associacions de SLURM amb usuaris, clúster i comptes permetent aplicar d'una forma abstracte unes polítiques d'ús dels recursos. Aquesta forma és la que el seu nom indica: proporciona a l'administració la manera de treballar d'un banc, on es donen crèdits a persones, aquestes gasten els diners, en funció de l'ús se li donen interessos, etc. El crèdit es tradueixen en recursos del clúster.

  Crec que aquesta pot ser una bona forma per facilitar la tasca als administradors del servei de càlcul de CIMNE i també per proporcionar als usuaris una abstracció de l'actual sistema de prioritats amb “FairShare”.

  Veure la presentació disponible a Sched MD:

  “*SLURM Bank*, Jimmy Tang and Paddy Doyle, Trinity College, Dublin, October 2012”

– Ample de banda com a recurs

  Un dels problemes que se'ns ha presentat és que els nodes M600 no escalen correctament en codis que fan un ús intensiu de memòria. Per aquest motiu hem limitat a 4 els processos que corren sobre els nodes. Això crea un problema i és que no s'aprofiten sempre tots els recursos, fent així el factor d'amortització del servei més petit. Per això volem analitzar el PFC fet per un alumne de la UPC sobre aquest tema, disponible a UPC Commons:

  Millora de la gestió de recursos a SLURM, Carlos Fenoy García, Juny 2010.

– Anàlisis de la diferència entre recursos reservats i recursos utilitzats

  Alguns usuaris abusen del fet de poder reservar molta quantitat de recursos encara que tinguin baixes prioritats. Això fa que per exemple, en un cas concret, el node Pez013 estigués ocupat per un sol procés i amb 5GiB de RAM al RSS, havent reservat 1 sol nucli però els 48GB disponibles.

  En el futur seria interessant recollir aquest tipus d'estadística per poder fer un estudi de com s'utilitzen els recursos i com solucionar el problema.

– Més eines

  Actualment el clúster disposa d'alguns debuggers i eines de programació, però seria ideal analitzar quines altres més avançades es poden fer servir per codi paral·lel i instal·lar-les al clúster proporcionant sempre una documentació adecuada. Per exemple, Totalview, Paraver, Tau, etc.

  Al LLNL ens mostren quines fan servir ells i ens pot servir com a orientació inicial:

  https://computing.llnl.gov/?set=code&page=software_tools

– Servei de classes als usuaris

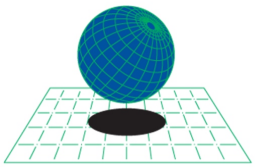
  Un cop entès el funcionament de tot el sistema seria molt interessant dedicar recursos a disposar d'administradors de sistema i uns altres a que algú muntés un servei d'ajuda a la programació en HPC. D'aquesta manera conseguiriem millorar el factor d'amortització del servei de càlcul i a més ajudaríem als investigadors a crear codis més eficients, traduint-se tot això amb resultats per l'empresa.

– E-mails a Sbatch

  En aquesta ocasió es tracta d'un petit detall interessant pels usuaris. El fet és que s'hauria de configurar SLURM per permetre enviar un correu als usuaris quan el seu treball enviat a cues canviés d'estat, així estarien informats en tot moment de com evoluciona la tasca.

– Checkpoint & Restart

  Aquesta funcionalitat implementada a SLURM però que no hem configurat per no creure necessària al principi, ha resultat que després d'alguns moments hagués servit molt a alguns usuaris. El fet és que alguns processos s'han parat degut a talls d'electricitat externs a CIMNE i els usuaris han perdut dies de càlculs. El fet seria estudiar quina freqüència de checkpoints fer als treballs i definir un espai per emmagatzemar-los, evitant així la pèrdua d'aquests en reinicis o problemes.



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Bibliografia

[1] Pàgina web de CIMNE - *http://www.cimne.com*

[2] Pàgina web d'Iberdrola, disposa d'informació sobre les tarifes de l'energia elèctrica del 2012.

https://www.iberdrola.es/webibd/corporativa/iberdrola?IDPAG=ESWEBCLIHOGASEINFLEGELE

[3] Wikipedia, ITIL v3 - *http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library*

[4] Redmine – *http://www.redmine.org*

[5] Project Manager + SVN, CIMNE *https://svn.cimne.upc.edu/p/cluster*

[6] Socomec Innovative Power Solutions - *http://www.socomec.com*

[7] Dell Poweredge M1000e - *http://www.dell.com/us/enterprise/p/poweredge-m1000e/pd*

[8] KVM Switches - *http://avocent.com*

[9] iKVM Dell - *http://support.dell.com/support/edocs/software/smdrac3/cmc/cmc1.0/en/ug/html/ikvm.htm*

[10] Manual del propietari Dell PowerEdge M1000e - *http://support.dell.com/support/edocs/systems/pem/12GOM/m1000e/sp/m1000eomsp.zip*

[11] Switch Infiniband Cisco M SFS7000E DDR 4x Installation guide - *http://www.cisco.com/en/US/docs/server_nw_virtual/m_sfs7000e/installation/note/sfs7000e.html*

[12] Dell Poweredge M610 Technical Guide - *http://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/server-poweredge-m610-tech-guidebook.pdf*

[13] Broadcom iSCSI Offload - *http://www.dell.com/Downloads/Global/Power/ps4q08-20080208-Broadcom.pdf*

[14] Blades Made Simple - *http://bladesmadesimple.com/*

[15] SAS 6ir - *http://accessories.us.dell.com/sna/products/controllers/productdetail.aspx?c=us&l=en&s=bsd&cs=ama&sku=a3064871*

[16] Dell Poweredge M600 - *http://www.dell.com/us/dfb/p/poweredge-m600/pd*

[17] Dell Poweredge M605 - *http://www.dell.com/us/dfb/p/poweredge-m605/pd*

[18] iSCSI Offload Engine - *http://www.dell.com/content/topics/global.aspx/power/en/broadcom_iscsi_offload_engine?c=us&l=en*

[19] FB-DIMM, How does it work? - *http://www.hardwaresecrets.com/article/266*

[20] FB-DIMM & Intel Processors - *http://www.intel.com/cd/channel/reseller/asmo-na/eng/products/server/processors/250634.htm*

[21] FB-DIMM - *http://en.wikipedia.org/wiki/Fully_Buffered_DIMM*

[22] Sun Ultra 40 M2 Manual - *http://docs.oracle.com/cd/E19127-01/ultra40.ws/820-0123-13/index.html*

[23] Sun XFire 4600 M2 Manual - *http://docs.oracle.com/cd/E19121-01/sf.x4600/819-4342-18/html/z40007e81010242.html*

[24] AMD Opteron 885 - *http://shop.amd.com/us/All/Detail/Processor/OSA885CCWOF*

[25] AMD Opteron 2214 - *http://products.amd.com/(S(f4uagsi5p14ll445uhlhvz45))/pages/OpteronCPUDetail.aspx?id=314*

[26] Infiniband Wikipedia - *http://en.wikipedia.org/wiki/InfiniBand*

[27] RDMA Wikipedia - *http://en.wikipedia.org/wiki/RDMA*

[28] Gestió Switch CISCO Infiniband - *http://www.cisco.com/en/US/docs/server_nw_virtual/m_sfs7000e/software/supplementary/note/7000e_sw.pdf*

[29] Presentació Infiniband CISCO -

http://www.cisco.com/web/DK/assets/docs/presentations/tech2007/Preso_Ji_Lim_session3_PDF.pdf

[30] PCISig - *http://www.pcisig.com/news_room/faqs/pcie2.0_faq/*

[31] Dell WhitePaper, PCI Express Technology - *http://content.dell.com/us/en/corp/d/business~solutions~whitepapers~en/Documents~wp-2004_pciexpress.pdf.aspx?redirect=1*

[32] Comparing Fabric Technologies - *http://www.datastorageconnection.com/doc.mvc/Comparing-Fabric-Technologies-InfiniBand-Arch-0001*

[33] Beowulf Cluster - *http://en.wikipedia.org/wiki/Beowulf_cluster*

[34] Seymour Cray - *http://en.wikipedia.org/wiki/Seymour_Cray*

[35] Supercomputador - *http://en.wikipedia.org/wiki/Supercomputer*

[36] Grid Computing - *http://en.wikipedia.org/wiki/Grid_computing#Definitions*

[37] Intel QPI - *http://en.wikipedia.org/wiki/Intel_QuickPath_Interconnect*

[38] Arquitectura QPI - *http://www.qdpma.com/systemarchitecture/systemarchitecture_qpi.html*

[39] NUMA and Node Interleaving - *http://frankdenneman.nl/2010/12/node-interleaving-enable-or-disable/*

[40] Dell Optimal Bios Settings for HPC - *http://i.dell.com/sites/content/business/solutions/whitepapers/ja/Documents/HPC_Dell_11g_BIOS_Options_jp.pdf*

[41] The Architecture of the Nehalem Processor and Nehalem-EP SMP Platforms – *Michael E. Thomadakis, Texas A&M University - 17/03/2011 - http://sc.tamu.edu/systems/eos/nehalem.pdf*

[42] Wikipedia, ccNuma - *http://en.wikipedia.org/wiki/Non-Uniform_Memory_Access#Cache_coherent_NUMA_.28ccNUMA.29*

[43] HyperTransport Consortium - *http://www.hypertransport.org/*

[44] Top500 Statistics - *http://i.top500.org/stats*

[45] RedHat Purchasing Entrerprise Linux - *https://www.redhat.com/resourcelibrary/articles/articles-red-hat-enterprise-linux-purchasing-guide*

[46] Scientic Linux - *http://www.scientificlinux.org/*

[47] RHEL 6 pam.d docs - *https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Migration_Planning_Guide/ch07s05.html*

[48] Dell Linux wiki and MD Storage Manager -

http://linux.dell.com/wiki/index.php/Products/HA/DellRedHatHALinuxCluster/Storage/PowerVault_MD3000/Software#RDAC_Multi-Path_Proxy_Driver

[49] Scientific Linux Mirror Repos - *http://www.scientificlinux.org/download/mirroring/mirror.rsync*

[50] RedHat Docs - *https://access.redhat.com/knowledge/docs/Red_Hat_Enterprise_Linux/*

[51] Firewall Iptables basics - *http://www.sysresccd.org/Sysresccd-Networking-EN-Iptables-and-netfilter-load-balancing-using-connmark*

[52] 25 Most freq. Used linux iptables rules - *http://www.thegeekstuff.com/2011/06/iptables-rules-examples/*

[53] Munge - *http://code.google.com/p/munge/wiki/InstallationGuide*

[54] Sched MD Slurm docs - *http://www.schedmd.com/slurmdocs/*

[55] Lawrence Livermore National Laboratory, Parallel computing -

https://computing.llnl.gov/tutorials/parallel_comp/

[56] MPI Forum - *http://www.mpi-forum.org/*

[57] IBM Infiniband test HowTo - *http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/index.jsp?topic=%2Fliaai%2Fhpcsuse%2Ftestib.htm*

[58] Wikipedia Intel Tick-tock - *http://en.wikipedia.org/wiki/Intel_Tick-Tock*

[59] Wikipedia, Micro-arquitectura Intel Core - *http://en.wikipedia.org/w/index.php?title=File:Intel_Core2_arch.svg&page=1*

[60] Intel Core Architecture - *http://docs.notur.no/uit/archive/HPCiA07/hpcia07-documents/intel-tools-tutorial/1.%20Core%20Architecture.pdf*

[61] QDPMA Website, 5400 Architecture - *http://www.qdpma.com/systemarchitecture/SystemArchitecture_2011Q3.html*

[62] Intel 5000P datasheet - *http://ark.intel.com/products/27746/Intel-5000P-Memory-Controller*

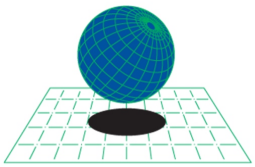
[63] AMD Codenames explained - *http://www.brightsideofnews.com/print/2009/5/1/amd-opteron2c-phenom-codenames-explained.aspx*

[64] RealWorldTech, Analisis AMD Barcelona vs Core2 - *http://www.realworldtech.com/barcelona/2/*

[65] Bit-tech, Core2 vs Nehalem - *http://www.bit-tech.net/hardware/cpus/2008/11/03/intel-core-i7-nehalem-architecture-dive/5*

[66] UPC Sirena - *http://www.upc.edu/sirena/*

[67] CarbonTrust, kWh to kg of CO^2 - *www.carbontrust.com/resources/reports/advice/conversion-factors*



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Annex 1
Documents interns

A.1.1 Taula de documents interns

Els documents interns d'aquesta secció es classifiquen en dos tipus en funció de la seva llicència o permís de distribució. Hi ha alguns documents com factures, especificacions tècniques, pressupostos, etc. que només té autorització a l'accés el personal intern de CIMNE, aquests seran classificats com a privats i es troben a la unitat de xarxa interna a CIMNE:

\\masterdisc\sistemes\Documentacion_variada\Proyectos\Servei_de_calcul\doc\privat

Els documents públics es poden visitar, tant des de la unitat de xarxa interna

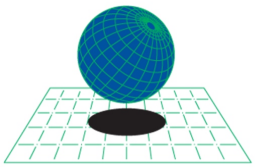
\\masterdisc\sistemes\Documentacion_variada\Proyectos\Servei_de_calcul\doc\public

com des de la pàgina web:

https://web.cimne.upc.edu/groups/sistemes/index.php?dir=Servicios+de+calculo

#Ref	*Nom del document*	*Públic*	*Descripció*
[doc1]	2008-01-valoracio proposta v0.xls	✓	Taula amb la valoració realitzada respecte els concursos del cluster adquirit l'any 2008.
[doc2]	Desglose de precios DELL QUOTE_ES_REL_PUBL_13639980.5_2008-04-08.pdf	✗	Factura del cluster adquirit l'any 2008.
[doc3]	informe_de_instalacion_cimne_27_Oct_2008.pdf	✗	Informe d'instal·lació de *Linalco Consulting S.L*. que ens va deixar al haver realitzat la instal·lació del cluster al 2008.
[doc4]	GestiónRecursosCálculo.pdf	✓	Presentació Café CIMNE de la nova infraestructura de càlcul. 2008. Per Miguel Pasenau.
[doc5]	CalculoCompartidoPresentacion.pdf	✓	Presentació de l'estructura d'emmagatzamemament de xarxa i servidors de càlcul disponibles. 2008. Per Miguel Pasenau.
[doc6]	Rack-Poweredge4210-InstallationGuide	✓	Manual d'instal·lació del rack que allotja el cluster.
[doc7]	Electricidad Cluste pe_m1000e_selection_whitepaper.pdf	✓	Guia completa sobre la configuració elèctrica del rack per un chassis M1000e.
[doc8]	Preso_Ji_Lim_session3_PDF.pdf	✓	Presentació resumida però molt completa sobre la tecnologia Infiniband. Font: [29]
[doc9]	wp-2004_pciexpress.pdf	✓	WhitePaper de Dell on explica en detall els busos PCI i PCIexpress. Font: [31]
[doc10]	Sun Fire X4600 Datasheet.pdf	✓	Especificacions del servidor Sun Fire X4600

#Ref	Nom del document	Públic	Descripció
[doc11]	Sun Fire X4600 Datasheet.pdf	✓	Especificacions del servidor Sun Fire X4600
[doc12]	Sun Ultra 40 Datasheet.pdf	✓	Especificacions del servidor Sun Ultra 40
[doc13]	Sun Ultra 40 M2 Datasheet.pdf	✓	Especificacions del servidor Sun Ultra 40M2
[doc14]	Flynns-Taxonomy.pdf	✓	Taxonomia de Flynn i explicació de diferents arquitectures
[doc15]	Cisco Infiniband WhitePaper.pdf	✓	WhitePaper de CISCO explicant aspectes de l'Infiniband.
[doc16]	ReestructuracióSAN-MD300i-0.mpeg	✓	Vídeo de la primera part de la reestructuració del SAN.
[doc17]	ReestructuracióSAN-MD300i-1.mpeg	✓	Vídeo de la segona part de la reestructuració del SAN.
[doc18]	Dell PowerConnect 54xx UserGuide.pdf	✓	Manual d'usuari del switch Dell PowerConnect 54xx



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Annex 2
Glossari de termes

A.2.1 Glossari de termes i vocabulari

Moltes de les paraules que a continuació es descriuen són originals de l'anglès. No es tradueixen per facilitar-ne la comprensió i recerca d'informació.

chassis
: Estructura que suporta la inserció de n servidors de tipus blade i que proporciona una forma compacta de col·locació d'aquests a dins del rack. A més facilita la connexió entre diferents servidors, la distribució d'electricitat, la ventilació del conjunt i la gestió de tot el conjunt mitjançant una sola consola d'accés. Un exemple de chassis utilitzat en aquest document és el chassis “PowerEdge M1000e”.

clúster de càlcul
: Conjunt d'ordinadors connectats mitjançant algun tipus de xarxa i que tenen com a finalitat unificar recursos per el càlcul computacional. També existeixen clusters d'alta disponibilitat que repliquen diferents serveis, però no es tracten en aquest projecte.

blade
: Tipus de forma de servidor de mides reduïdes que té la característica de poder ser integrat a dins d'un chassis. Pretén optimitzar l'espai usat, l'energia consumida i a més està pensat reduir el nombre de cables necessari per el seu funcionament. És l'antònim a un servidor enrackable o a un de torre.

rack
: Armari metàl·lic destinat a la col·locació de servidors i al seu aprovisionament ordenat de cables d'electricitat i comunicacions. Solen estar dissenyats per facilitar el flux d'aire per el seu interior i tenen unes mides i tipus d'enclavaments normalitzats. Habitualment tenen una amplada estàndard de 600mm i un fons de 800 o 1000mm. S'hi poden allotjar servidors de tipus blade, enrackables de 2U, 1U, etc.

switch
: Commutador de xarxa. Pot ser programable o no programable.

framework
: Defineix una estructura conceptual i tecnològica normalment amb mòduls, classes, biblioteques i/o eines de software que proporciona una base per treballar i permet desenvolupar un projecte de software de forma metodològica.

yum
: Yellowdog Updater, Modified. Sistema de gestió de paquets de distribucions GNU/Linux amb empaquetament de tipus RPM.

crontab
: Referència que es fa a la taula de tasques planificades del cron. Cron és un dimoni típic dels sistemes Unix que executa processos en segon pla a intervals de temps determinats.

Infiniband

Xarxa d'altres prestacions d'estàndard IEEE 802.3.z que utilitza un bus de dades bi-direccional i en topologia “switched fabric”, cada node connectat a tot altre mitjançant 1 o més enllaços punt a punt. El tipus utilitzat en aquest projecte és el DDR 4x que pot arribar a un màxim de 20Gbps en topologia Fat Tree.

SM

Sigles de Subnet Manager, és un dimoni responsable de la gestió de la xarxa en un entorn Infiniband. És indispensable i realitza tasques com encaminament, gestió de ports, canvis d'estat, particions, camins, etc.

iSCSI

Protocol de comunicació que treballa sobre IP i que permet l'enviament de paquets orientats a l'emmagatzemament de xarxa de forma més eficient que el transport sobre TCP.

iSOE

iSCSI Offload Engine. Tecnología de dispositiu hardware que es pot col·locar a la placa base dels servidors i que ofereix la possibilitat de descarregar a la CPU de gestionar el tràfic de xarxa iSCSI.

RDMA

Remote Direct Memory Access. Tecnologia que permet a una interfície de xarxa accedir a la memòria d'una aplicació i fer transferències de dades des d'aquell espai de memòria a un altre espai de memòria d'una altra aplicació d'un altre node, sense requerir intervenció del sistema operatiu evitant així canvis de context, còpies redundants, etc. S'utilitza com a base de l'Infiniband.

HBA

Host Bus Adapter. Element de comunicacions com pot ser una targeta de xarxa que disposa d'un hardware específic que descarrega a la CPU de realitzar les tasques de gestió del tràfic que encamina. Per exemple en una Ethernet es diferència d'una targeta convencional en que porta un dispositiu TCP Offload Engine (TOE).

HCA

Host Channel Adapter. En Infiniband és el nom que se li dona a les targetes controladores o HBAs.

beowulf

Tipus de clúster de càlcul en que s'utilitzen ordinadors personals o de baix cost i xarxes de comunicació ethernet enlloc de hardware especialitzat o propietari. Són models de memòria distribuïda i fan servir pas de missatges MPI o semblant.

PDU

Power Distribution Unit, és un lladre de corrent elèctric que es caracteritzar per ser utilitzat en entorns de centres de processament de dades i comunicacions.

PSU

Power Supply Unit, terme anglès que es refereix a la font d'alimentació.

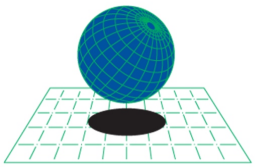
SAI
Sistema d'alimentació ininterrompuda. És un element elèctric que disposa de bateries i que donada una corrent d'entrada permet mantenir una corrent estable a la seva sortida. També té la capacitat de treballar sense alimentació d'entrada el temps suficient perquè un administrador del CPD o un software automatitzat pugui realitzar les accions necessàries als servidors, per exemple apagar-los.

KVM
Dispositiu que permet realitzar el control de diversos elements mitjançant un únic monitor, teclat i ratolí.

Service Tag
Nombre de sèrie atorgat als equips Dell que permet obtenir suport per el producte en qüestió i veure les seves característiques i garantia:

http://www.dell.com/support/troubleshooting/us/en/555/ProductSelector

out-band-management
Terme utilitzat per referir-se a la gestió que es fa d'un dispositiu des de una xarxa externa. El terme contrari és in-band-management que es referiria a una gestió amb una connexió directa.



CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Annex 3
Informació de referència

A.3.1 Característiques del servei de càlcul

Hardware

Clúster de càlcul:	Dell™ PowerEdge™ m1000e chassis
Node d'accés:	1 PowerEdge™ M600
Nodes de càlcul:	15 nodes PowerEdge™ M600, M605 i M610
Emmagatzemament:	Dell™ PowerVault™ MD3000i iSCSI SAN Array
	10TiB Raid 5 muntat per NFS
Comunicacions:	Ethernet 1GB Dell PassThrough
	Infiniband Switch Cisco M SFS7000E DDR 4x
Total Nuclis càlcul:	132
Toral Memòria RAM:	368 GB
Total TFLOPS:	1,240

Software

Sistema Operatiu:	Scientific Linux 6.1 (RedHat EL 6.1)
Kernel:	2.6.32-220 x86_64
Gestor de recursos:	SLURM 2.4.3
Planificador de tasques:	SLURM 2.4.3
GCC:	4.4.5 20110214 (RedHat 4.4.5-6)
Open MPI:	1.6.2-1
Mvapich2:	1.4-5
Intel MPI:	4.0.3.008
Intel Composer XE:	2011 SP1.6.233

Característiques dels nodes del servei de càlcul

	#nodes	Model node	Sockets	Micro-arquitectura	Processador	Freq.	Nuclis	Fils / cpu	RAM
Node màster	1	Dell™ PowerEdge™ M600	2	Harpetown (Penryn)	Intel®Xeon® E5410	2.33GHz	4	4	32GB
Nodes E5410	10	Dell™ PowerEdge™ M600	2	Harpetown (Penryn)	Intel®Xeon® E5410	2.33GHz	4	4	16GB
Nodes AMD2356	2	Dell™ PowerEdge™ M605	2	Barcelona (K10)	AMD Opteron™ 2356	2.30GHz	4	8	16GB
Nodes E5645	3	Dell™ PowerEdge™ M610	2	Westmere-EP (Nehalem)	Intel®Xeon® E5645	2.40GHz	6	12	48GB

Taula 18: Característiques dels nodes

Particions del planificador de tasques SLURM

Partition name	*Time limit*	*#nodes*	*Node list*	*Model*	*Cores per node*	*Memory per node*	*Hyper-threading*	*Intended usage*
Main	90 days	8	Pez001 to Pez008	M600	Limited to 4	16GB	OFF	General usage, single thread, MPI and OpenMP
Short	1 day	2	Pez009 to Pez010	M600	Limited to 4	16GB	OFF	1 day short time computations, single thread and OpenMP
AMD2356	90 days	2	Pez011 to Pez012	M605	8	32GB	OFF	Parallel OpenMP problems and general scalability tests
XeonE5645	90 days	3	Pez013 to Pez015	M610	12	48GB	OFF	Very high parallel performance requeriments

Taula 19: Particions de SLURM

Característiques específiques dels processadors

```
architecture:   x86_64
      vendor:         GenuineIntel
      processor-name: Intel(R) Xeon(R) CPU E5645  @ 2.40GHz
      model:          Family 6, Model 44, Stepping 2
      frequency:      2393 MHz
      total number of (logical) cores in system: 12
      number of (logical) cores per package: 6
        supported features: FPU MMX SSE SSE2 SSE3 SSSE3 SSE4.1 SSE4.2
  POPCNT CX8 CX16 MONITOR NX CPUID MTRR
                          TSC: 21 cycles latency
                          CLFLUSH: 64 Byte clflush-linesize
      Level1 Cache:
        - separated Instruction and Data Caches
        - 32768 Bytes I-Cache, 4-way set-associative
        - 32768 Bytes D-Cache, 8-way set-associative
        - per CPU
        - 64 Byte Cachelines
      Level2 Cache:
        - unified Cache
        - 262144 Bytes, 8-way set-associative
        - per CPU
        - 64 Byte Cachelines
      Level3 Cache:
        - unified Cache
        - 12582912 Bytes, 16-way set-associative
        - shared between 6 CPU(s)
        - 64 Byte Cachelines
      supported pagesizes: 4 KiByte, 2 MiByte
      virtual address length:  48 bits
      physical address length: 40 bits
      Level1 ITLB:
        128 entries for 4 KiByte pages, 4-way set associative
        7 entries for 2 MiByte pages, fully associative
      Level1 DTLB:
        64 entries for 4 KiByte pages, 4-way set associative
        32 entries for 2 MiByte pages, 4-way set associative
      Level2 TLB (code and data):
        512 entries for 4 KiByte pages, 4-way set associative
```

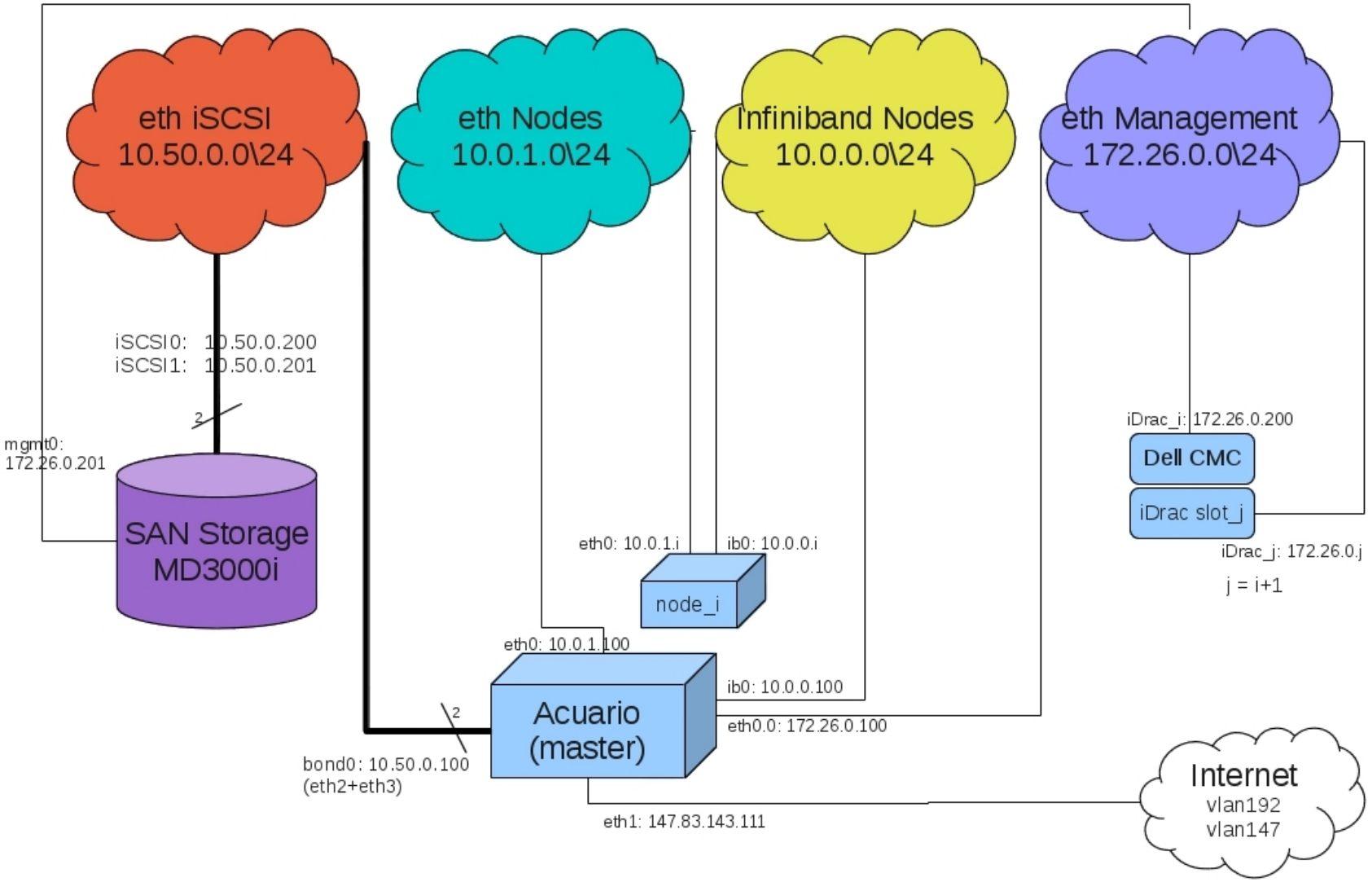
```
architecture:   x86_64
    vendor:         AuthenticAMD
    processor-name: Quad-Core AMD Opteron(tm) Processor 2356
    model:          Family 16, Model 2, Stepping 3
    frequency:      2299 MHz
    total number of (logical) cores in system: 8
    number of (logical) cores per package: 4
    supported features: FPU MMX MMX_EXT SSE SSE2 SSE3 SSE4A POPCNT CX8
CX16 FREQ_SCALING MONITOR NX CPUID HAP MTRR
                        TSC: 65 cycles latency
                        CLFLUSH: 64 Byte clflush-linesize
    Level1 Cache:
      - separated Instruction and Data Caches
      - 65536 Bytes I-Cache, 2-way set-associative
      - 65536 Bytes D-Cache, 2-way set-associative
      - per CPU
      - 64 Byte Cachelines
    Level2 Cache:
      - unified Cache
      - 524288 Bytes, 16-way set-associative
      - per CPU
      - 64 Byte Cachelines
    Level3 Cache:
      - unified Cache
      - 2097152 Bytes, 32-way set-associative
      - shared between 4 CPU(s)
      - 64 Byte Cachelines
    supported pagesizes: 4 KiByte, 2 MiByte, 1 GiByte
    virtual address length:  48 bits
    physical address length: 48 bits
    Level1 ITLB:
      32 entries for 4 KiByte pages, fully associative
      16 entries for 2 MiByte pages, fully associative
    Level1 DTLB:
      48 entries for 4 KiByte pages, fully associative
      48 entries for 2 MiByte pages, fully associative
      48 entries for 1 GiByte pages, fully associative
    Level2 ITLB:
      512 entries for 4 KiByte pages, 4-way set associative
    Level2 DTLB:
      512 entries for 4 KiByte pages, 4-way set associative
      128 entries for 2 MiByte pages, 2-way set associative
```

```
architecture:   x86_64
    vendor:         GenuineIntel
    processor-name: Intel(R) Xeon(R) CPU           E5410  @ 2.33GHz
    model:          Family 6, Model 23, Stepping 6
    frequency:      2333 MHz
    total number of (logical) cores in system: 8
    number of (logical) cores per package: 4
     supported features: FPU MMX SSE SSE2 SSE3 SSSE3 SSE4.1 CX8 CX16
FREQ_SCALING MONITOR NX CPUID MTRR
                        TSC: 32 cycles latency
                        CLFLUSH: 64 Byte clflush-linesize
    Level1 Cache:
      - separated Instruction and Data Caches
      - 32768 Bytes I-Cache, 8-way set-associative
      - 32768 Bytes D-Cache, 8-way set-associative
      - per CPU
      - 64 Byte Cachelines
    Level2 Cache:
      - unified Cache
      - 6291456 Bytes, 24-way set-associative
      - shared between 2 CPU(s)
      - 64 Byte Cachelines
    supported pagesizes: 4 KiByte, 2 MiByte
    virtual address length:  48 bits
    physical address length: 38 bits
    Level1 ITLB:
      128 entries for 4 KiByte pages, 4-way set associative
      8 entries for 2 MiByte pages, 4-way set associative
    Level1 DTLB:
      256 entries for 4 KiByte pages, 4-way set associative
```

A.3.2 Service Tags

Slot	*Hostname*	*Model*	*Service Tag*
001	acuario	PowerEdge M600	74T4Q3J
002	pez001	PowerEdge M600	H8T4Q3J
003	pez002	PowerEdge M600	FCT4Q3J
004	pez003	PowerEdge M600	7DT4Q3J
005	pez004	PowerEdge M600	HCT4Q3J
006	pez005	PowerEdge M600	1DT4Q3J
007	pez006	PowerEdge M600	DCT4Q3J
008	pez007	PowerEdge M600	5DT4Q3J
009	pez008	PowerEdge M600	8CT4Q3J
010	pez009	PowerEdge M600	BCT4Q3J
011	pez010	PowerEdge M600	3CT4Q3J
012	pez011	PowerEdge M605	9JT4Q3J
013	pez012	PowerEdge M605	BJT4Q3J
014	pez013	PowerEdge M610	9P4315J
015	pez014	PowerEdge M610	BP4315J
016	pez015	PowerEdge M610	8P4315J
Chassis	-	PowerEdge M1000e	7JT4Q3J
Dell CMC	-		
SAN MD3000i	-	PowerVault MD3000i	
Switch 5424	-	PowerConnect 5424	

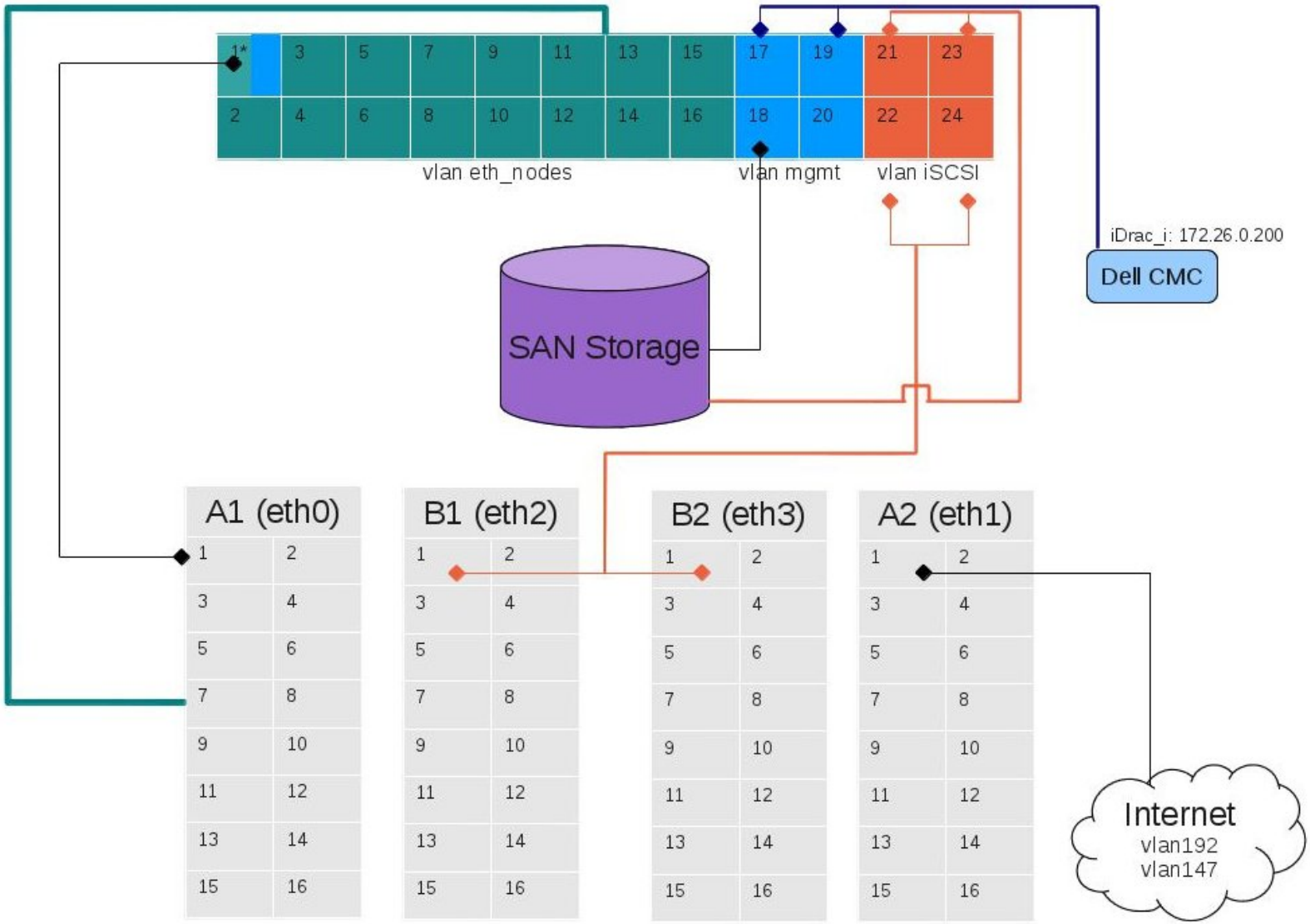
A.3.3 Esquema de la xarxa – VLANs



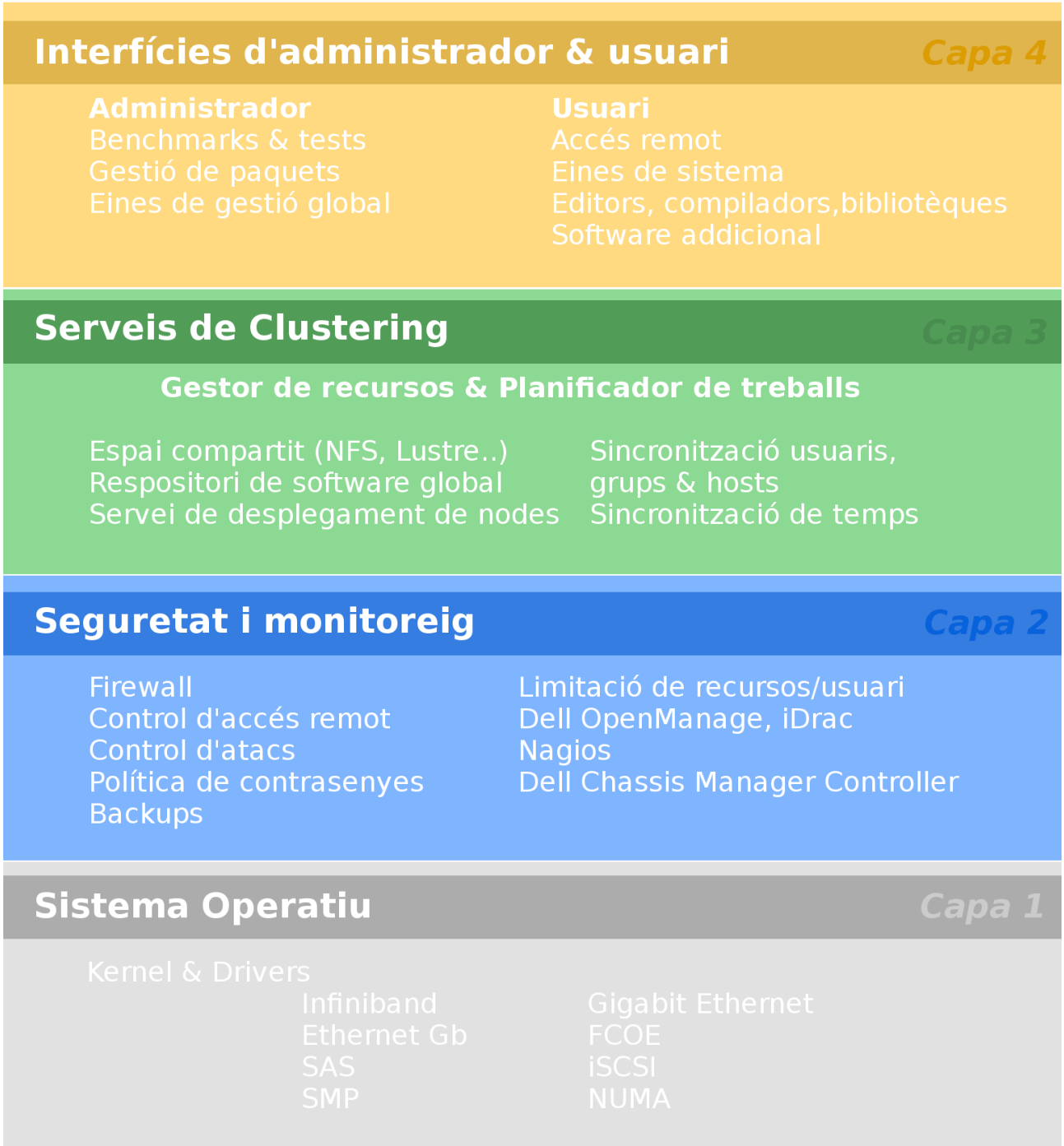
A.3.4 Adreçament IP

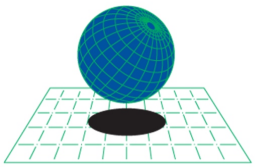
Slot	*Hostname*	*Infiniband Nodes*	*eth Nodes*	*eth Management*	*eth iSCSI*	*Internet*	*iDrac*
001	acuario	10.0.0.100	10.0.1.100	172.26.0.100	10.50.0.100	147.83.143.111	172.26.0.1
002	pez001	10.0.0.1	10.0.1.1				172.26.0.2
003	pez002	10.0.0.2	10.0.1.2				172.26.0.3
004	pez003	10.0.0.3	10.0.1.3				172.26.0.4
005	pez004	10.0.0.4	10.0.1.4				172.26.0.5
006	pez005	10.0.0.5	10.0.1.5				172.26.0.6
007	pez006	10.0.0.6	10.0.1.6				172.26.0.7
008	pez007	10.0.0.7	10.0.1.7				172.26.0.8
009	pez008	10.0.0.8	10.0.1.8				172.26.0.9
010	pez009	10.0.0.9	10.0.1.9				172.26.0.10
011	pez010	10.0.0.10	10.0.1.10				172.26.0.11
012	pez011	10.0.0.11	10.0.1.11				172.26.0.12
013	pez012	10.0.0.12	10.0.1.12				172.26.0.13
014	pez013	10.0.0.13	10.0.1.13				172.26.0.14
015	pez014	10.0.0.14	10.0.1.14				172.26.0.15
016	pez015	10.0.0.15	10.0.1.15				172.26.0.16
Dell CMC				172.26.0.200			
MD3000i				172.26.0.201	10.50.0.200/201		
Switch 5424				172.26.0.202			

A.3.5 Cablejat de xarxa



A.3.6 Pila de capes d'un clúster de memòria distribuïda





CIMNE

Centre Internacional de Mètodes Numèrics en Enginyeria

Annex 4
Fitxers de configuració

A.4.1 /tftpboot/pxelinux.cfg/default

```
# Fitxer de configuració del menu de PXE BOOT
# Opcions rellevants:
# ONTIMEOUT : Indica quin LABEL triar passats TIMEOUT milisegons.
# LABEL : Defineix diferents opcions d'engegat
# "biosdevname=0" : En nodes Dell nous, la bios permet nombrar les
# interficies pel nom
# relacionat a la posició fisica real. Nosaltres volem el sistema classic
# perque potser no tots els nodes en el futur seran Dell.
DEFAULT menu.c32
PROMPT 0
TIMEOUT 100

#ONTIMEOUT scientific
ONTIMEOUT local

NOESCAPE 1
ALLOWOPTIONS 0
MENU TITLE Acuario PXE Menu

#Definicio del menú
LABEL local
MENU LABEL Inicia desde disc dur
LOCALBOOT 0

LABEL scientific
MENU LABEL Scientific 6.1 Instalacio Node
KERNEL images/scientific/6.1/x86_64/vmlinuz
APPEND      initrd=images/scientific/6.1/x86_64/initrd.img      biosdevname=0
ramdisk_size=100000 ksdevice=eth0 ip=dhcp ks=nfs:10.0.1.100:/kickstart/ks.cfg

LABEL rescue
MENU LABEL Scientific Linux 6.1 Rescue
KERNEL images/scientific/6.1/x86_64/vmlinuz
APPEND      initrd=images/scientific/6.1/x86_64/initrd.img      biosdevname=0
ramdisk_size=10000            text            ksdevice=eth0            rescue
ks=nfs:10.0.1.100:/kickstart/rescue.cfg
```

A.4.2 /etc/dhcp/dhcpd.conf

```
[root@acuario tftpboot]# cat /etc/dhcp/dhcpd.conf
# dhcpd.conf
#
# Configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
option domain-name "acuario.cimne.upc.edu";
option domain-name-servers 147.83.143.133;

default-lease-time 21600;
max-lease-time 43200;

# Use this to enble / disable dynamic dns updates globally.
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

subnet 10.0.1.0 netmask 255.255.255.0 {
         #range 10.0.1.1 10.0.1.15; No podem posar ips que declarem a hosts
fixes
        #perquè sinos aquestes es poden donar a altres clients que no siguin
les fixes.
       range 10.0.1.16 10.0.1.20;
       option routers 10.0.1.100;
       option subnet-mask 255.255.255.0;
       option ntp-servers 10.0.1.100;

       allow booting;
       allow bootp;
       next-server 10.0.0.100;
       filename "/pxelinux.0";

       host pez001 {
                hardware ethernet 00:1E:C9:CD:2A:DC;
          fixed-address 10.0.1.1;
       }
       host pez002 {
            hardware ethernet 00:1E:C9:CD:3B:D5;
          fixed-address 10.0.1.2;
       }
      host pez003 {
                hardware ethernet 00:1E:C9:CD:35:4B;
          fixed-address 10.0.1.3;
       }
       host pez004 {
                hardware ethernet 00:1E:C9:CD:3B:A9;
          fixed-address 10.0.1.4;
       }
```

```
        host pez005 {
                 hardware ethernet 00:1E:C9:CD:31:2A;
           fixed-address 10.0.1.5;
        }
        host pez006 {
                 hardware ethernet 00:1E:C9:CD:38:81;
           fixed-address 10.0.1.6;
        }
        host pez007 {
                 hardware ethernet 00:1E:C9:CD:38:A5;
           fixed-address 10.0.1.7;
        }
        host pez008 {
                 hardware ethernet 00:1E:C9:CD:26:8A;
           fixed-address 10.0.1.8;
        }
        host pez009 {
                 hardware ethernet 00:1E:C9:CD:35:4F;
           fixed-address 10.0.1.9;
        }
        host pez010 {
                 hardware ethernet 00:1E:C9:CD:31:2E;
           fixed-address 10.0.1.10;
        }
        host pez011 {
                 hardware ethernet 00:18:8B:FF:9D:5A;
           fixed-address 10.0.1.11;
        }
        host pez012 {
                 hardware ethernet 00:1D:09:FC:C0:6C;
           fixed-address 10.0.1.12;
        }
        host pez013 {
                 hardware ethernet 5C:26:0A:FE:2C:6C;
           fixed-address 10.0.1.13;
        }

        host pez014 {
                 hardware ethernet 5C:26:0A:FE:2C:E0;
           fixed-address 10.0.1.14;
        }

        host pez015 {
                 hardware ethernet 5C:26:0A:FE:2C:4C;
           fixed-address 10.0.1.15;
        }
}
```

A.4.3 /kickstart/ks.cfg

```
[root@acuario tftpboot]# cat /kickstart/ks.cfg
#################################################
########### INSTALLATION PARAMETERS #############
#################################################

#platform=x86, AMD64, o Intel EM64T
#version=DEVEL
# Firewall configuration
firewall --disabled
# Install OS instead of upgrade
install
# Use NFS installation media
nfs --server=10.0.1.100 --dir=/repo/scientific/6.1/x86_64/os/
# Root password
rootpw --iscrypted $1$C/aZaSzp$WpA8J6dt7gbFq67JAyvNJ.
# Network information
# Beware here: Due to a bug, is possible that the NetworkManager doesn't get
# an IP address. See:
# https://bugzilla.redhat.com/show_bug.cgi?id=663820
# https://bugzilla.redhat.com/show_bug.cgi?id=669019
network  --bootproto=dhcp --device=eth0 --onboot=on
#network --bootproto=bootp --device=eth0 --onboot=on
# System authorization information
auth  --useshadow  --passalgo=sha512
# Use text mode install
text
# System keyboard
keyboard es
# System language
lang en_US
# SELinux configuration
selinux --disabled
# Do not configure the X Window System
skipx
# Installation logging level
logging --level=info
# Reboot after installation
reboot
# System timezone
timezone  Europe/Madrid
# System bootloader configuration
bootloader --location=mbr
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all --initlabel
# Disk partitioning information
part swap --fstype="swap" --ondisk=sda --recommended
part / --fstype="ext4" --grow --ondisk=sda --size=1
```

```
################################################
############# PRE INSTALL SCRIPTS ##############
################################################
%pre
#!/bin/sh
# Mount /repo located at the eth address of master node to get access to the
repository of packages
mkdir /repo
mount    -t    nfs    -o    ro,nolock,hard,udp,vers=3,rsize=32768,wsize=32768
10.0.1.100:/repo /repo
%end

################################################
############ POST INSTALL SCRIPTS ##############
################################################

%post --log=/root/instalacio-ks-acuario.log
#!/bin/sh

# Stage 1
# First mount configuration directory and /repo to get current OS updates
# These directories are mounted through the ethernet IP address of master
node
mkdir /kickstart
mount    -t    nfs    -o    ro,nolock,hard,udp,vers=3,rsize=32768,wsize=32768
10.0.1.100:/kickstart /kickstart

mkdir /repo
mount    -t    nfs    -o    ro,nolock,hard,udp,vers=3,rsize=32768,wsize=32768
10.0.1.100:/repo /repo

## Stage 2
## In the configuration directory we can find the common operating system
config files,
## so we copy it to the new installation, adapt it to the name of the new
node,
## and perform some other tasks.

# YUM repositories
# Set YUM and disable automatic system upgrades
rm -f /etc/yum.repos.d/*
cp -vf /kickstart/acuario/etc/yum.repos.d/* /etc/yum.repos.d/

#chkconfig yum-updatesd off
#chkconfig yum-updateonboot off
#chkconfig yum-cron off
yum -y remove yum-autoupdate
yum -y update

# Create the global file system mount point
mkdir /globalfs

# Setup fstab mounts
cat  << xxEOFxx >> /etc/fstab
eth_acuario:/repo /repo nfs defaults 0 0
eth_acuario:/home /home nfs defaults 0 0
eth_acuario:/globalfs /globalfs nfs defaults 0 0
xxEOFxx
```

```
# Enable network time synchronization and adjust the hardware clock
cp -vf /kickstart/acuario/etc/ntp.conf /etc/
chkconfig --level 2345 ntpd on
/usr/sbin/ntpdate -u -b -s 10.0.1.100
/sbin/hwclock --utc --systohc

# Set up the network
# Copy the common hosts file
cp -vf /kickstart/acuario/etc/hosts /etc/

# Enable networking and Infiniband
chkconfig network on
chkconfig rdma on

# Obtain the node name from their IP address given by dhcp at installation
start
# The IP address must be given to the interface eth0
NODE_I=`/sbin/ifconfig eth0 | sed -n '/inet addr:.*/{s/.*inet addr://; s/
.*//;p}' | cut -d"." -f 4`
HOSTNAME=`cat /etc/hosts | grep -w "10.0.0.$NODE_I" | cut -f2 | cut -d" "
-f1`

# Copy the hostname, eth0, ib0 and ib1 network config files and set them for
this particular node
rm /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network
sed   -e   s/xxx/"$HOSTNAME"/g   /kickstart/acuario/etc/sysconfig/network   >
/etc/sysconfig/network
sed     -e     s/xxx/"$NODE_I"/g     /kickstart/acuario/etc/sysconfig/network-
scripts/ifcfg-eth0 > /etc/sysconfig/network-scripts/ifcfg-eth0
sed     -e     s/xxx/"$NODE_I"/g     /kickstart/acuario/etc/sysconfig/network-
scripts/ifcfg-ib0 > /etc/sysconfig/network-scripts/ifcfg-ib0
cp                  /kickstart/acuario/etc/sysconfig/network-scripts/ifcfg-ib1
/etc/sysconfig/network-scripts/

## Stage 3
## We must prepare the system to run within the cluster infraestructure.

# Enable NIS
# It allows the synchronization of passwd, shadow, group and hosts files from
the master node
nisdomainname cimne.upc.edu
echo -e "\nNISDOMAIN=cimne.upc.edu" >> /etc/sysconfig/network
chkconfig ypbind on
cp -Rpf /kickstart/acuario/etc/yp.conf /etc/
cp -Rpf /kickstart/acuario/etc/host.conf /etc/
cp -Rpf /kickstart/acuario/etc/nsswitch.conf /etc/

# Enable root SSH auto-login. The private key is stored in /root/.ssh in the
# master node
mkdir /root/.ssh
cp -vf /kickstart/acuario/authorized_keys /root/.ssh/
chmod 700 /root/.ssh
chmod 750 /root/.ssh/*

# Setup the SSH server
# For example, we deny the SSH access to users different than root
cp -Rpf /kickstart/acuario/etc/ssh/sshd_config /etc/ssh/
chmod 400 /etc/ssh/sshd_config
```

```
# Setup Slurm and Munge
# Slurm is the scheduler and resource manager of Acuario Cluster.
# We install the client daemon here, slurmd
#  Munge is the daemon that provides an authentication system to Slurm
resource manager, we install the client munged

yum -y install slurm slurm-munge slurm-plugins munge
cp -Rpf /kickstart/acuario/etc/munge/munge.key /etc/munge/

# Link to the cluster common config files and setup dirs
ln -s /globalfs/etc/slurm/slurm.conf /etc/slurm/slurm.conf
ln -s /globalfs/etc/sysconfig/slurm /etc/sysconfig/slurm
mkdir /var/log/slurm/
chown -R munge:munge /etc/munge/
chmod 0700 /etc/munge/
chmod 400 /etc/munge/munge.key

chkconfig munge on
chkconfig slurm on

# Install and setup Ganglia client daemon, for cluster monitoring
yum -y install ganglia-gmond
rm /etc/ganglia/gmond.conf
ln -s /globalfs/etc/ganglia/gmond.conf /etc/ganglia/gmond.conf
chkconfig gmond on

## Stage 4
## Install libraries, compilers, analyzers and other useful tools

# Install mpi
yum -y install openmpi

# Install valgrind AFTER MPI in order not to get openmpi installed outside
the cluster-packages repo.
yum -y install valgrind valgrind-openmpi

# Setup the software shared directory across the cluster
# There you can find Intel Compiler, Matlab, etc.
rm -rf /opt
ln -s /globalfs/opt /opt

# Unmount config dir
umount /kickstart
rm -rf /kickstart
%end
```

```
################################################
############ PACKAGES TO INSTALL  ###############
################################################

## All this packages will be installed from the Official Repositories.
## We include drivers, compilers and libraries needed for the cluster.
%packages
@base
@infiniband
@network-file-system-client
@network-tools
@performance
infiniband-diags
perftest
rdma
librdmacm
librdmacm-utils
sed
ntp
ypbind
gcc
blas
lapack
papi
make
cmake
gcc
libXt
libXmu
tcsh
-b43-fwcutter
-bridge-utils
-cryptsetup-luks
-eject
-ethtool
-fprintd-pam
-hunspell
-libipathverbs
-mdadm
-mlocate
-mtr
-openswan
-oprofile
-pcmciautils
-pinfo
-plymouth
-pm-utils
-rdate
-rfkill
-setuptool
-sl-release-notes
-smartmontools
-sos
-system-config-firewall-tui
-system-config-network-tui
-wireless-tools
-words
%end
```

A.4.4 /etc/slurmdbd.conf

```
#
# Example slurmdbd.conf file.
#
# See the slurmdbd.conf man page for more information.
#
# Archive info
#ArchiveDir="/tmp"
ArchiveEvents=yes
ArchiveJobs=yes
ArchiveSteps=no
ArchiveSuspend=no
#ArchiveScript=
#JobPurge=12
#StepPurge=1
#
# Authentication info
AuthType=auth/munge
#AuthInfo=/var/run/munge/munge.socket.2
#
# slurmDBD info
DbdAddr=acuario
DbdHost=acuario
DbdPort=6819
SlurmUser=slurm
#MessageTimeout=300
#Debug Level,min 0 to 9 max
DebugLevel=0
#DefaultQOS=normal,standby
LogFile=/var/log/slurm/slurmdbd.log
PidFile=/var/run/slurmdbd.pid
PluginDir=/usr/local/lib/slurm
#PrivateData=accounts,users,usage,jobs
#TrackWCKey=yes
#
# Database info
StorageType=accounting_storage/mysql
StorageHost=localhost
StoragePort=3306
StoragePass=3kuartsD15
StorageUser=slurm
StorageLoc=slurm_acct_db

PurgeEventAfter=6month
PurgeJobAfter=12month
PurgeStepAfter=1month
PurgeSuspendAfter=1month
```

A.4.5 /etc/slurm/slurm.conf - /globalfs/etc/slurm.conf

```
#########################################################
# slurm.conf
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.
# Felip Moll Marquès - 2011
# CIMNE
#########################################################

#################################
########## GENERAL ##############
#################################

######### DAEMON #######
ControlMachine=acuario
#ControlAddr=
#BackupController=
#BackupAddr=
SlurmctldPidFile=/var/run/slurmctld.pid
SlurmctldPort=6817
SlurmdPidFile=/var/run/slurmd.pid
SlurmdPort=6818
SlurmdSpoolDir=/var/spool/slurmd/
SlurmUser=slurm
StateSaveLocation=/tmp/slurm.state
#PluginDir=
#PlugStackConfig=
#MailProg=/bin/mail
#TmpFs=/tmp
#SallocDefaultCommand="$SHELL"
#GresTypes=

##Daemon logging
#DebugFlags=
SlurmctldDebug=3
SlurmdDebug=3
SlurmctldLogFile=/var/log/slurm/slurmctld.log
SlurmdLogFile=/var/log/slurm/slurmd.%h.log

######## SECURITY ##########
AuthType=auth/munge
#Si activem caché de grups, haurem de fer scontrol reconfig quan canviem
passwords o grups del sistema. Avegades
#cal reiniciar el servei. Sembla algun bug, el desactivem per ara.
CacheGroups=0
CryptoType=crypto/munge
#JobCredentialPrivateKey=
#JobCredentialPublicCertificate=
#GroupUpdateForce=0
#GroupUpdateTime=600
#PrivateData=jobs
#UsePAM=1

######### JOBS ###########
#FirstJobId=1
#MaxJobCount=5000
#JobFileAppend=0
```

```
#JobRequeue=1
#JobSubmitPlugins=1
#DisableRootJobs=NO
#EnforcePartLimits=NO
#KillOnBadExit=0
#CheckpointType=checkpoint/none
#JobCheckpointDir=/var/slurm/checkpoint
#Licenses=foo*4,bar
#UnkillableStepProgram=

#Scripts to run as root after or before performing some actions
#Prolog=/usr/global/bin/prolog.sh
Epilog=/etc/slurm/slurm.epilog.clean
#PrologSlurmctld=
#TaskEpilog=
#TaskProlog=
#SrunEpilog=
#SrunProlog=

######## LIMITS ##########
#MaxTasksPerNode=128
#PropagatePrioProcess=0
#PropagateResourceLimits=
#PropagateResourceLimitsExcept=MEMLOCK,AS
PropagateResourceLimitsExcept=AS
###### MISC PLUGINS ########
MpiDefault=none
MpiParams=ports=10000-20000

ProctrackType=proctrack/linuxproc
SwitchType=switch/none

TaskPlugin=task/affinity
TaskPluginParam=Sched

#TopologyPlugin=topology/tree
#TreeWidth=

##############################
########### TIMERS ###########
##############################
#BatchStartTimeout=10
#CompleteWait=0
#EpilogMsgTime=2000
#GetEnvTimeout=2
#HealthCheckInterval=0
#HealthCheckProgram=
InactiveLimit=0
KillWait=30
#MessageTimeout=10
#ResvOverRun=0
MinJobAge=300
#OverTimeLimit=0
SlurmctldTimeout=120
SlurmdTimeout=300
#UnkillableStepTimeout=60
#VSizeFactor=0
Waittime=0
```

```
###############################
########### SCHEDULING #######
###############################
#SLURM can either base its scheduling decisions upon the node configuration
defined in slurm.conf
#or what each node actually returns as available resources.
#Set its value to zero in order to use the resources actually found on each
node,
#but with a higher overhead for scheduling.

#SlurmSchedLogFile=
#SlurmSchedLogLevel=
#SchedulerRootFilter=1

FastSchedule=1
SchedulerPort=7321
SelectType=select/cons_res
SelectTypeParameters=CR_Core_Memory
DefMemPerCPU=1024
MaxMemPerCPU=0

##GANG SCHEDULING - Uncomment to activate
##Not recommended to have nodes in multiple partitions
##Remember to comment backfill or preemption
#DefMemPerCPU=0
#MaxMemPerCPU=0
#SchedulerTimeSlice=90
#SchedulerType=sched/builtin
#PreemptMode=GANG
#PreemptType=preempt/partition_prio

##Backfill SCHEDULING - Uncomment to activate
##Remember to comment GANG or PREEMPTION
SchedulerType=sched/backfill
PreemptMode=OFF

##Preemption - Uncomment to activate
#PreemptMode=
#PreemptType=preempt/partition_prio

###############################
########### JOB PRIORITY #####
###############################
#See https://computing.llnl.gov/linux/slurm/priority_multifactor.html
PriorityType=priority/multifactor
#2 week half-life
PriorityDecayHalfLife=14-0
#If we don't use Decay Half-life we must reset the calculus periodically
#PriorityUsageResetPeriod=
#Recalc every 10 minuts
PriorityCalcPeriod=10
#The smaller the job, the greater its job size priority (and priority).
PriorityFavorSmall=YES
#The job's age factor reaches 1.0 after waiting in the queue for 3 days
PriorityMaxAge=3-0
#This next group determines the weighting of each of the
#components of the Multi-factor plugin.
#The default value for each of the following is 1.
PriorityWeightAge=100000
```

```
PriorityWeightFairshare=60000
PriorityWeightJobSize=10000
PriorityWeightPartition=1000
#Don't use the qos factor
PriorityWeightQOS=0

################################
######## ACCOUNTING ##########
################################
AccountingStorageEnforce=associations
AccountingStorageHost=acuario
AccountingStoragePort=6819
AccountingStorageType=accounting_storage/slurmdbd
AccountingStorageUser=slurm
ClusterName=Acuario
#AccountingStorageLoc=/var/log/slurm/job-accounting-storage.log
#AccountingStoragePass=
#JobCompHost=
#JobCompLoc=/var/log/slurm/completed-jobs.log
#JobCompPass=
#JobCompPort=
#JobCompType=jobcomp/filetxt
#JobCompUser=
JobAcctGatherFrequency=30
JobAcctGatherType=jobacct_gather/linux
#TrackWCKey=no

#####################################################
##### POWER SAVE SUPPORT FOR IDLE NODES (optional)#
#####################################################
#SuspendProgram=                        #SuspendExcNodes=
#ResumeProgram=                         #SuspendExcParts=
#SuspendTimeout=                        #SuspendRate=
#ResumeTimeout=                         #SuspendTime=
#ResumeRate=

#################################################
###### COMPUTE NODES and PARTITIONS #############
#################################################
ReturnToService=2

NodeName=pez001 Sockets=2 CoresPerSocket=2 ThreadsPerCore=1 RealMemory=32109
NodeName=pez[002-010] Sockets=2 CoresPerSocket=2 ThreadsPerCore=1 RealMemory=15950
NodeName=pez[011-012] Sockets=2 CoresPerSocket=4 ThreadsPerCore=1 RealMemory=32169
NodeName=pez[013-015] Sockets=2 CoresPerSocket=6 ThreadsPerCore=1 RealMemory=48251


PartitionName=Main  Nodes=pez[001-008] Default=YES MaxTime=90-0 Priority=0 State=UP
PartitionName=Short Nodes=pez009,pez010 Default=NO MaxTime=1-0 Priority=0 State=UP
PartitionName=AMD2356 Nodes=pez011,pez012 Default=NO MaxTime=90-0 Priority=0 State=UP
PartitionName=XeonE5645 Nodes=pez0[13-15] Default=NO MaxTime=90-0 Priority=0 State=UP
```

A.4.6 /etc/ganglia/gmond.conf del node màster

Posem només les parts rellevants:

```
globals {
  daemonize = yes
  setuid = yes
  user = nobody
  debug_level = 0
  max_udp_msg_len = 1472
  mute = no
  deaf = no
  allow_extra_data = yes
  host_dmax = 0
  host_tmax = 60 /*secs, timeout max */
  cleanup_threshold = 300 /*secs */
  gexec = no
  send_metadata_interval = 20 /*secs */
}

cluster {
  name = "Cluster Acuario"
  owner = "CIMNE"
  latlong = "N41.38 E2.11"
  url = "http://www.cimne.com"
}

host {
  location = "CIMNE - BCN CPD - 108A"
}

udp_send_channel {
#  bind_hostname = yes
  host = 10.0.1.100
  port = 8600
  ttl = 1
}

udp_recv_channel {
  port = 8600
  bind = 10.0.1.100
}

tcp_accept_channel {
  port = 8600
}

#udp_recv_channel {
#  port = 6343
#}

#sflow {
# udp_port = 6343
# accept_vm_metrics = no
#}

-------------------------------8<------------------------------
```

A.4.7 /etc/ganglia/gmond.conf dels nodes esclau

Posem només les parts rellevants:

```
globals {
  daemonize = yes
  setuid = yes
  user = nobody
  debug_level = 0
  max_udp_msg_len = 1472
  mute = no
  deaf = no
  allow_extra_data = yes
  host_dmax = 86400
  host_tmax = 20 /*secs */
  cleanup_threshold = 300 /*secs */
  gexec = no
  send_metadata_interval = 0 /*secs */
}

cluster {
  name = "Cluster Acuario"
  owner = "CIMNE"
  latlong = "N41.38 E2.11"
  url = "http://www.cimne.com"
}

host {
  location = "CIMNE - BCN CPD - 108A"
}

udp_send_channel {
#  bind_hostname = yes
  host = 10.0.1.100
  port = 8600
  ttl = 1
}

#udp_recv_channel {
#  mcast_join = 239.1.1.1
#  port = 8600
#}

tcp_accept_channel {
  port = 8600
}

#udp_recv_channel {
#  port = 6343
#}

#sflow {
# udp_port = 6343
# accept_vm_metrics = no
#}

------------------------------8<------------------------------
```

A.4.8 /etc/ganglia/gmetad.conf

Ometem tots els comentaris:

```
data_source "Cluster Acuario" 10.0.1.100:8600
RRAs "RRA:AVERAGE:0.5:1:105408"
xml_port 8651
interactive_port 8652
case_sensitive_hostnames 1
```

A.4.9 /etc/modulefiles/impi-4.0.3

```
#%Module
##
## Mòdul per carregar les variables d'entorn de Intel MPI
##
##
## Simula la comanda: source /opt/intel/impi/4.0.3/bin64/mpivars.[sh,csh]
##
## Felip Moll - 14/10/2011
##
proc ModulesHelp { } {
      puts stderr "Sets up the paths you need to use Intel MPI Compilers and
libraries v.4.0.3"
}

module-whatis    Intel MPI implementation, compilers and libraries
conflict      openmpi-x86_64  mvapich2-x86_64    mvapich-psm-x86_64    mvapich-
x86_64

prepend-path       MANPATH      /opt/intel/impi/4.0.3/man

eval set [ array get env HOME ]
setenv             INTEL_LICENSE_FILE          $HOME/intel/licenses

prepend-path       LD_LIBRARY_PATH       /opt/intel/impi/4.0.3/intel64/lib
prepend-path       PATH                  /opt/intel/impi/4.0.3/intel64/bin
setenv             I_MPI_ROOT            /opt/intel/impi/4.0.3
setenv             I_MPI_PMI_LIBRARY           /usr/lib64/libpmi.so
setenv             I_MPI_FABRICS               shm:dapl
setenv             I_MPI_CC              icc
setenv             I_MPI_FC              ifort
setenv             I_MPI_CXX             icpc
setenv             I_MPI_F77             ifort
setenv             I_MPI_F90             ifort
```

A.4.10 /etc/modulefiles/intelcc-12.1.0

```
#%Module
##
## Mòdul per carregar les variables d'entorn del Intel Composer XE
##
## Simula la comanda: source /opt/intel/bin/compilervars.[sh,csh] intel64
## Felip Moll - 14/10/2011
##
proc ModulesHelp { } {
      puts stderr "Sets up the paths you need to use Intel Compilers and libraries
v.12.1.0 including mkl"
}

module-whatisLoads Intel Composer XE (C++, C and Fortran compilers, MKL library)

prepend-path  MANPATH      /opt/intel/composer_xe_2011_sp1.6.233/man/en_US

eval set [ array get env HOME ]
setenv        INTEL_LICENSE_FILE      $HOME/intel/licenses
prepend-path  INTEL_LICENSE_FILE      /opt/intel/licenses
prepend-path  INTEL_LICENSE_FILE      /opt/intel/composer_xe_2011_sp1.6.233/licenses

setenv        IPPROOT                 /opt/intel/composer_xe_2011_sp1.6.233/ipp
setenv        MKLROOT                 /opt/intel/composer_xe_2011_sp1.6.233/mkl
prepend-path  LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/compiler/lib/intel64
prepend-path  LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/ipp/../compiler/lib/intel64
prepend-path  LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/ipp/lib/intel64
prepend-path  LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/compiler/lib/intel64
prepend-path  LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/mkl/lib/intel64
prepend-path  LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/tbb/lib/intel64//cc4.1.0_libc2.4_kernel2.6.16.2
1

setenv        FPATH
/opt/intel/composer_xe_2011_sp1.6.233/mkl/include
setenv        CPATH
/opt/intel/composer_xe_2011_sp1.6.233/mkl/include:/opt/intel/composer_xe_2011_sp1.6.2
33/tbb/include

prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/compiler/lib/intel64
prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/ipp/../compiler/lib/intel64
prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/ipp/lib/intel64
prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/compiler/lib/intel64
prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/mkl/lib/intel64
prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/tbb/lib/intel64//cc4.1.0_libc2.4_kernel2.6.16.2
1
prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/debugger/lib/intel64
prepend-path  LD_LIBRARY_PATH
/opt/intel/composer_xe_2011_sp1.6.233/mpirt/lib/intel64

prepend-path  NLSPATH
/opt/intel/composer_xe_2011_sp1.6.233/compiler/lib/intel64/locale/%l_%t/%N/
```

```
prepend-path  NLSPATH
/opt/intel/composer_xe_2011_sp1.6.233/ipp/lib/intel64/locale/%l_%t/%N
prepend-path  NLSPATH
/opt/intel/composer_xe_2011_sp1.6.233/mkl/lib/intel64/locale/%l_%t/%N
prepend-path  NLSPATH
/opt/intel/composer_xe_2011_sp1.6.233/debugger/intel64/locale/%l_%t/%N

prepend-path  PATH
/opt/intel/composer_xe_2011_sp1.6.233/mpirt/bin/intel64
prepend-path  PATH
/opt/intel/composer_xe_2011_sp1.6.233/bin/intel64

setenv        TBBROOT                 /opt/intel/composer_xe_2011_sp1.6.233/tbb
prepend-path  INCLUDE
/opt/intel/composer_xe_2011_sp1.6.233/ipp/include
prepend-path  INCLUDE
/opt/intel/composer_xe_2011_sp1.6.233/mkl/include
```

A.4.11 /etc/modulefiles/cmake-2.8.8

```
#%Module
##
## Mòdul per carregar les variables d'entorn de CMake 2.8.8
##
## Felip Moll - 27/06/2012
##
proc ModulesHelp { } {
     puts stderr "Sets up the paths you need to use the CMake 2.8.8 tool
     instead of 2.6.4 installed on the system"
}

module-whatis     Loads path environment variables to CMake 2.8.8

prepend-path       MANPATH      /globalfs/opt/cmake-2.8.8/man/
prepend-path       PATH       /globalfs/opt/cmake-2.8.8/bin/
```

A.4.12 /etc/modulefiles/matlab-2011b

```
#%Module 1.0
#
#  Matlab module for use with 'environment-modules' package:
#

prepend-path            PATH        /opt/MATLAB/R2011b/links
```

Index alfabètic

O

P

Q

R

S

T

U

V

W

X

Y

/